



**Pedro Filipe Vieira  
Rito**

**Recetor SDR para Comunicações DSRC**





**Pedro Filipe Vieira  
Rito**

## **Recetor SDR para Comunicações DSRC**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor João Nuno Pimentel da Silva Matos e do Doutor Arnaldo Silva Rodrigues de Oliveira, Professores do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro





**o júri / the jury**

presidente / president

**Professor Doutor Nuno Miguel Gonçalves Borges de Carvalho**

Professor Associado com agregação da Universidade de Aveiro

vogais / examiners committee

**Professor Doutor José Carlos dos Santos Alves**

Professor Associado do Departamento de Engenharia Eletrotécnica e Computadores da Faculdade de Engenharia da Universidade do Porto

**Professor Doutor João Nuno Pimentel da Silva Matos**

Professor Associado da Universidade de Aveiro (Orientador)

**Professor Doutor Arnaldo Silva Rodrigues de Oliveira**

Professor Auxiliar da Universidade de Aveiro (Coorientador)



## **agradecimentos / acknowledgements**

A realização desta dissertação não seria possível sem a ajuda e suporte de algumas pessoas.

Em primeiro lugar, quero agradecer aos meus orientadores, pela sua ajuda, coordenação e alguma paciência durante este último ano.

Ao Ricardo Matos Abreu por ter ajudado a intensificar os meus conhecimentos em Sistemas de Comunicações, por ter dado suporte a problemas pontuais e por ter seguido o projeto tão de perto, ajuda essencial para a implementação deste Recetor.

A todos os meus amigos que conheci na Universidade, que me acompanharam durante todo o curso. Ao lado deles, a passagem por esta fase académica foi acima de tudo bastante divertida.

Aos meus amigos de Fátima por todos os fins-de-semanas e férias que passámos juntos. Momentos importantes que contribuíram para a minha evolução como pessoa, carregando e refrescando as minhas energias.

À minha namorada, pela presença na minha vida, ao seu apoio, ajuda, força, carinho, recomendações e influências no meu dia-a-dia. Com ela, tudo que consegui até agora foi conseguido com mais facilidade e motivação.

Por fim, o agradecimento mais importante, à minha Família. A minha enorme gratidão pela educação e atenção que me deram. Bases essenciais para os meus sonhos e metas até agora concretizados.

Obrigado por tudo!



## Palavras-chave

Comunicações Veiculares, Cobrança Eletrónica de Taxas (EFC), DSRC, SDR, *Costas loop*

## Resumo

Nos últimos anos, um dos focos de investigação tem sido a conceção e inclusão de sistemas inteligentes no tráfego rodoviário. Chamados *Intelligent Transportation Systems* (ITS), têm como finalidade minimizar o tempo perdido em congestionamentos, consumo de energia e danos ambientais, humanos e materiais, tentando assim contribuir para a sustentabilidade do setor. Um dos exemplos mais conhecidos é a cobrança eletrónica de taxas, *Electronic Fee Collection* (EFC). O EFC usa comunicações sem fios de curto alcance, normalmente definidas por *Dedicated Short-Range Communications* (DSRC).

Atualmente, existem algumas questões que precisam de ser ultrapassadas para ser possível o desenvolvimento efetivo destes sistemas. Por estarmos perante sistemas que estão em constante evolução são necessárias novas tecnologias de implementação dos terminais que tornem possível a sua atualização frequente. A melhor solução para satisfazer este tipo de requisitos é recorrer à utilização do conceito de *Software Defined Radio* (SDR), isto é, tratar os sinais de rádio tanto quanto possível no domínio digital e, eventualmente, com recurso a *software*.

Esta dissertação tem como objetivo o desenvolvimento de parte de um recetor BPSK para DSRC recorrendo aos conceitos de SDR. O recetor de rádio inclui a conversão de analógico para digital dos sinais recebidos e de seguida a desmodulação em FPGA. Para isso, é feita inicialmente uma apresentação dos *standards* que envolvem as comunicações DSRC a nível europeu, tal como as vantagens da aplicação de SDR neste projeto.

Com vista à conceptualização do projeto, são estudadas e simuladas algumas das arquiteturas mais importantes na receção de sinais BPSK. Partindo da análise das vantagens e desvantagens de cada uma destas, é escolhida a melhor para utilizar na implementação. Essa arquitetura utiliza o *Costas loop*.

Seguidamente, é efetuada a apresentação de forma detalhada dos dois principais módulos implementados para o recetor: a construção da placa de amplificação, filtragem e conversão analógico-digital dos sinais recebidos, e a realização do desmodulador BPSK em FPGA.

Numa última fase, recorrendo a alguns testes de validação e avaliação do sistema, são apresentados vários resultados que ilustram o funcionamento do sistema. Conclui-se que o recetor é capaz de desmodular os sinais recebidos. A dissertação termina com algumas sugestões com vista à evolução do equipamento no futuro.



**Keywords**

Vehicular Communications, Electronic Fee Collection (EFC), DSRC, SDR, Costas loop

**Abstract**

Recently, a focus by the research teams has been the conception and inclusion of intelligent systems in road traffic. Called Intelligent Transportation Systems (ITS), they are intended to minimize the time lost in congestion, energy consumption and environmental, human and material damage, thus trying to contribute to the sustainability of the sector. One of the known examples is the Electronic Fee Collection (EFC). EFC uses Dedicated Short-Range Communications (DSRC).

Currently, there are some issues that need to be achieved to the effective development of these systems. Because we are dealing with systems that have constant evolution, new technologies are needed for design of these terminals that make possible to be frequently updated. The best solution to meet such requirements is to use Software Defined Radio (SDR), that is, handle the radio signals as much as possible by digital domain and, eventually, using software.

This dissertation aims to develop part of a BPSK receiver for DSRC using SDR. The radio receiver includes conversion of received signals from analog to digital and then the demodulation in FPGA. Therefore, it is initially made a presentation of communications involving DSRC standards at European level, such as the advantages of using SDR in this project.

For the conceptualization of the project to be implemented, they are studied and simulated some of the most important architectures in the reception of BPSK signals. Analyzing the advantages and disadvantages of each one, the demodulation method chosen for the implementation was the Costas loop.

Next, it is made a detailed presentation of the two main modules of receiver implementation: the construction of the amplification, filtering and analog to digital conversion board, and the realization of BPSK demodulator in FPGA.

In a last phase, using validation and evaluation tests, several results are presented that illustrate the operation of the system. It was concluded that the receiver can demodulate the received signals. The dissertation ends with some suggestions for a maybe future evolution of the equipment.





# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>Acrónimos</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivos . . . . .	3
1.4 Estrutura da Dissertação . . . . .	3
<b>2 Conceitos Fundamentais</b>	<b>5</b>
2.1 Comunicações DSRC . . . . .	5
2.1.1 Introdução à Norma EN 12253 . . . . .	6
2.1.2 Espectro . . . . .	7
2.1.3 Modulação . . . . .	8
2.1.3.1 <i>Downlink</i> . . . . .	8
2.1.3.2 <i>Uplink</i> . . . . .	8
2.1.4 Preâmbulo do <i>Uplink</i> . . . . .	9
2.2 <i>Software Defined Radio</i> . . . . .	10
2.2.1 Configurações em SDR . . . . .	11
2.2.2 Digitalização em banda base . . . . .	11
2.2.3 Digitalização em IF . . . . .	12
2.2.4 Aplicação de SDR em DSRC . . . . .	13
2.3 Componentes usados em SDR . . . . .	13
2.3.1 <i>Analog-to-Digital Converters</i> (ADCs) . . . . .	13
2.3.1.1 <i>Pipelined</i> ADCs . . . . .	14
2.3.2 <i>Field Programmable Gate Arrays</i> (FPGAs) . . . . .	15
2.3.2.1 Arquitetura de FPGAs . . . . .	15
2.3.2.2 Comparação entre FPGAs e <i>Digital Signal Processor</i> (DSPs) . . . . .	15
2.3.2.3 Aplicação de FPGAs em SDR . . . . .	16
2.3.2.4 Implementação de Projetos em FPGAs . . . . .	16
2.3.2.5 Filtros <i>Finite Impulse Response</i> (FIR) em FPGAs . . . . .	17

<b>3</b>	<b>Modelação e Simulação do Recetor</b>	<b>19</b>
3.1	Introdução . . . . .	19
3.2	O BPSK . . . . .	19
3.3	Desmodulação BPSK . . . . .	21
3.3.1	Extração de portadora pelo quadrado do sinal recebido . . . . .	22
3.3.2	<i>Costas loop</i> . . . . .	24
3.4	Recuperação do relógio de transmissão em banda-base . . . . .	28
3.5	Esquema da Simulação . . . . .	30
3.6	Simulação de codificação NRZI e Modulação BPSK . . . . .	30
3.7	Introdução de ruído . . . . .	32
3.8	Andar de conversão analógico-digital . . . . .	33
3.9	Simulação desmodulador BPSK . . . . .	34
3.9.1	Filtro passa-banda . . . . .	34
3.9.2	Simulação da extração da portadora pelo quadrado do sinal recebido . . . . .	35
3.9.2.1	Filtro passa-banda a 4 MHz . . . . .	35
3.9.2.2	Saturar o sinal para criar uma onda quadrada . . . . .	36
3.9.2.3	Divisor de frequência por 2 . . . . .	36
3.9.2.4	Bloco de atraso “delay” . . . . .	37
3.9.2.5	Filtro passa-baixo para banda-base . . . . .	38
3.9.3	Simulação <i>Costas loop</i> . . . . .	38
3.9.3.1	Filtros passa-baixo (C e E) . . . . .	39
3.9.3.2	Correção do VCO através da realimentação . . . . .	39
3.9.3.3	Saída do <i>Costas loop</i> – sinal desmodulado . . . . .	40
3.9.4	Especificação do método de desmodulação a implementar . . . . .	41
3.10	Simulação do recuperador do relógio de transmissão em banda-base . . . . .	42
3.11	Resultados quantitativos da simulação . . . . .	44
<b>4</b>	<b>Implementação do Recetor</b>	<b>45</b>
4.1	Esquema do Recetor . . . . .	45
4.2	Front-end existente de 5.8 GHz . . . . .	47
4.3	Andar de Amplificação, Filtragem e Conversão AD . . . . .	50
4.3.1	Amplificador e filtro passa-banda . . . . .	51
4.3.2	Amplificador diferencial . . . . .	53
4.3.3	ADC . . . . .	54
4.3.3.1	Seleção da gama de amplitudes de tensão da entrada da ADC . . . . .	55
4.3.3.2	Seleção das entradas digitais de controlo da ADC . . . . .	56
4.4	Implementação do Desmodulador na FPGA . . . . .	57
4.4.1	Esquema geral do desmodulador . . . . .	57
4.4.2	Atraso de 90° no sinal em quadratura . . . . .	58
4.4.3	Saturação do sinal . . . . .	58
4.4.4	Filtros passa-banda . . . . .	59
4.4.5	Síntese do relógio de 16 MHz . . . . .	59
4.4.6	<i>Costas loop</i> . . . . .	60
4.4.6.1	<i>Numerically-controlled oscillator</i> (NCO) . . . . .	61
4.4.6.2	Filtro passa-baixo (C e E) . . . . .	62
4.4.6.3	Filtro passa-baixo do erro (G) . . . . .	63
4.4.6.4	Controlo do ganho da amplitude do erro à entrada do NCO . . . . .	64

4.4.7	Amostragem e Descodificação NRZI . . . . .	64
4.4.7.1	Integradores . . . . .	64
4.4.7.2	Cálculo de erro e amostragem . . . . .	65
4.4.7.3	Trigger . . . . .	66
4.4.7.4	Descodificação NRZI . . . . .	66
4.4.8	Placa de desenvolvimento da FPGA . . . . .	67
<b>5</b>	<b>Resultados</b>	<b>69</b>
5.1	Introdução . . . . .	69
5.2	Testes de avaliação e validação através do ChipScope . . . . .	69
5.2.1	Saturação e filtro passa-banda digital . . . . .	71
5.2.1.1	Sinal de entrada na FPGA – saída da ADC (A) . . . . .	71
5.2.1.2	Sinal saturado (B) . . . . .	72
5.2.1.3	Sinal depois do filtro passa-banda (C) . . . . .	72
5.2.2	<i>Costas loop</i> . . . . .	73
5.2.2.1	Sinal depois do filtro passa-baixo em fase (D) . . . . .	73
5.2.2.2	Sinal depois do filtro passa-baixo em quadratura (E) . . . . .	74
5.2.2.3	Erro depois do filtro passa-baixo dentro da malha (F) . . . . .	75
5.2.2.4	Sincronização entre o sinal recebido e a sub-portadora à saída do NCO (sin) . . . . .	75
5.2.3	Recuperação do relógio de transmissão em banda-base . . . . .	76
5.3	Testes de avaliação e validação através do osciloscópio e sem ruído adicionado . . . . .	77
5.3.1	Sinal de entrada na FPGA . . . . .	78
5.3.2	Sinal de correção do <i>Costas loop</i> (F) . . . . .	78
5.3.3	Sinal desmodulado à saída do <i>Costas loop</i> (D) . . . . .	79
5.3.4	Recuperação relógio banda-base . . . . .	79
5.3.5	Diagrama de olho do sinal desmodulado . . . . .	80
5.4	Testes de avaliação e validação através do osciloscópio e com ruído adicionado . . . . .	81
5.4.1	Sub-portadora recuperada (“sin”) . . . . .	81
5.4.2	Sinal de correção do <i>Costas loop</i> (F) . . . . .	82
5.4.3	Integrador e amostrador do sinal . . . . .	83
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>85</b>
6.1	Conclusões . . . . .	85
6.2	Trabalho Futuro . . . . .	86
	<b>Apêndices</b>	<b>87</b>
	<b>A Placa Construída (PCB e Esquema Elétrico)</b>	<b>89</b>
	<b>B Esquema da implementação na FPGA</b>	<b>93</b>
	<b>C Placa de Desenvolvimento Xilinx ML505</b>	<b>95</b>
	<b>Bibliografia</b>	<b>99</b>



# Lista de Figuras

2.1	Esquema de um sistema EFC. . . . .	6
2.2	Comunicação <i>Downlink</i> . . . . .	6
2.3	Comunicação <i>Uplink</i> . . . . .	7
2.4	Portadoras <i>Downlink</i> . . . . .	8
2.5	Portadoras <i>Uplink</i> . . . . .	8
2.6	Exemplo de aplicação de NRZI na codificação do <i>Uplink</i> . Sub-portadora a 1.5MHz. . . . .	9
2.7	Preâmbulo do <i>Uplink</i> . . . . .	9
2.8	A arquitetura SDR ideal. . . . .	11
2.9	Arquitetura simplificada de digitalização na banda base. . . . .	11
2.10	Arquitetura simplificada de digitalização em IF. . . . .	12
2.11	Arquiteturas de ADCs, aplicações, resolução e taxas de amostragem. [Kes05]	13
2.12	Etapas gerais ADCs <i>pipelined</i> . [Kes05] . . . . .	14
2.13	Diagrama temporal de uma ADC <i>pipelined</i> , de 12 <i>bits</i> , 65 MSPS, a AD9235. [Kes05] . . . . .	14
2.14	Exemplo da estrutura de FPGAs da Xilinx. . . . .	15
2.15	Fluxo de projeto em FPGAs. . . . .	16
2.16	Representação do atraso convencional da linha do filtro FIR [Xil10b] . . . . .	18
2.17	Diagrama temporal de um filtro FIR em FPGA. [Xil10b] . . . . .	18
3.1	Exemplo de forma de onda BPSK. No topo, o sinal em banda base, em baixo o sinal em BPSK. . . . .	20
3.2	PSD nas frequências positivas de um sinal BPSK. . . . .	20
3.3	Esquema de um detetor síncrono para sinais BPSK. . . . .	21
3.4	Esquema de um recuperador de portadora de sinal BPSK através de um circuito quadrador do sinal de entrada. . . . .	22
3.5	Esquema de um recetor BPSK com recuperação de portadora através de um circuito extrator. . . . .	23
3.6	Esquema completo de um recetor BPSK pelo método <i>Costas loop</i> . . . . .	24
3.7	Dependência da amplitude à entrada do VCO com o erro de fase do sinal recebido. . . . .	27
3.8	Esquema do recuperador de relógio dos símbolos transmitidos em banda-base. . . . .	28
3.9	Esboço do funcionamento do integrador de tempo finito. A azul, sinal em banda-base recebido à entrada do integrador. A vermelho, sinal integrado em tempo finito $T = 4\mu\text{seg}$ . . . . .	28
3.10	Diagrama de blocos da simulação usada. . . . .	30
3.11	Preâmbulo modulado a 2 MHz. . . . .	31

3.12	Espectro de frequência do sinal transmitido a 2 MHz. . . . .	31
3.13	Espectros de frequências dos sinais com e sem ruído. A vermelho, o sinal sem ruído. A azul, o sinal com ruído. . . . .	32
3.14	Resposta em frequência do filtro passa-banda do andar de conversão. . . . .	33
3.15	Espectro de frequências após o filtro passa-banda geral e ADC. A vermelho, o espectro depois do filtro e ADC. A azul, o espectro do sinal inicial com ruído. . . . .	33
3.16	Espectro de frequências após o filtro passa-banda geral e ADC com saturação na ADC. . . . .	34
3.17	Resposta em frequência do filtro passa-banda do desmodulador BPSK. . . . .	34
3.18	Diagrama de blocos do método de extração de portadora pelo quadrado do sinal recebido. . . . .	35
3.19	Resposta em frequência do filtro passa-banda de 4 MHz. . . . .	35
3.20	Espectro de frequências da subportadora a 4 MHz em forma de onda quadrada. . . . .	36
3.21	Exemplo de intervalo de tempo das sub-portadoras. A vermelho, sub-portadora a 4 MHz. A azul, sub-portadora depois de dividida, a 2 MHz. . . . .	37
3.22	Imagem do MATLAB do espectro de frequências da subportadora a 2 MHz em forma de onda quadrada. . . . .	37
3.23	Preâmbulo desmodulado, em banda-base. . . . .	38
3.24	Diagrama de blocos do método <i>Costas loop</i> . . . . .	38
3.25	Resposta em frequência do filtro passa-baixo em C e E. . . . .	39
3.26	Erro à entrada do VCO durante o período de preâmbulo. . . . .	40
3.27	Sincronização entre o sinal recebido e a sub-portadora à saída do VCO. A vermelho, sub-portadora gerada pelo VCO. A azul, sinal BPSK recebido. . . . .	40
3.28	Preâmbulo desmodulado, em banda-base, com referência à estabilização inicial do sistema. . . . .	41
3.29	Sobreposição de preâmbulos desmodulados com erros de fases diferentes na entrada do desmodulador. Verifica-se que o sinal fica sempre estável antes de antigrir o final do período de sub-portadora pura (aos $32\mu s$ ). . . . .	42
3.30	Diagrama de blocos do recuperador do relógio de transmissão em banda-base. . . . .	42
3.31	Preâmbulo desmodulado e integrado. São visíveis os três integradores em que o integrador central é o relativo à amostragem. Os pontos correspondem aos instantes em que ocorre a amostragem. . . . .	43
4.1	Arquitetura de digitalização em IF. . . . .	45
4.2	Esquema do recetor implementado. . . . .	46
4.3	Diagrama do front-end existente. . . . .	47
4.4	Comparação da probabilidade de erro de símbolos (BER) entre PSK e DPSK em função do SNR. . . . .	49
4.5	Diagrama de blocos do andar de amplificação, filtragem e conversão analógico-digital. . . . .	50
4.6	Esquema elétrico do circuito do amplificador e filtro passa-banda. . . . .	51
4.7	Comparação entre a simulação e o resultado retirado através do VNA da resposta em frequência do andar de amplificação e filtragem. . . . .	52
4.8	Esquema elétrico do circuito do amplificador diferencial. . . . .	53
4.9	Diagrama funcional ADC MAX1184. [MAX06] . . . . .	54
4.10	Esquema do desmodulador na FPGA. . . . .	57
4.11	Resposta em frequência dos dois filtros passa-banda implementados na FPGA. . . . .	59

4.12	Diagrama de blocos do <i>Costas loop</i> implementado na FPGA. . . . .	60
4.13	Sistema de blocos de um NCO. . . . .	61
4.14	Resposta em frequência do filtro passa-baixo usado em C e E do <i>Costas loop</i> . . . . .	62
4.15	Resposta em frequência do filtro passa-baixo usado em G do <i>Costas loop</i> . . . . .	63
4.16	Diagrama de blocos da amostragem e decodificação NRZI implementados na FPGA. . . . .	64
4.17	Esquema do funcionamento do bloco dos integradores implementado na FPGA. . . . .	65
4.18	Exemplo de funcionamento do bloco “trigger” implementado na FPGA. . . . .	66
4.19	Máquina de estados finitos de “Mealy” do bloco de decodificação NRZI implementado na FPGA. . . . .	66
5.1	Diagrama de blocos da montagem para testes através do ChipScope. . . . .	70
5.2	Diagrama da fase de saturação e de filtragem na FPGA. . . . .	71
5.3	Resultados dos sinais de entrada na FPGA a 1.5 MHz e 2 MHz durante o tempo de preâmbulo. . . . .	72
5.4	Resultados dos sinais a 1.5 MHz e 2 MHz saturados, durante o período de transição de oito 0’s no preâmbulo. . . . .	72
5.5	Resultados dos sinais a 1.5 MHz e 2 MHz depois do filtro passa-banda correspondente, durante o período de transição de oito 0’s no preâmbulo. . . . .	73
5.6	Diagrama do <i>Costas loop</i> na FPGA. . . . .	73
5.7	Resultados dos sinais transmitidos a 1.5 MHz e 2 MHz desmodulados depois do filtro passa-baixo em fase, durante o tempo do preâmbulo. . . . .	74
5.8	Resultados dos sinais transmitidos a 1.5 MHz e 2 MHz depois do filtro passa-baixo em quadratura, durante o tempo do preâmbulo. . . . .	74
5.9	Resultados da amplitude do erro à entrada do NCO de correção da sub-portadora gerada face aos sinais de entrada a 1.5 MHz e a 2 MHz, durante o tempo do preâmbulo. . . . .	75
5.10	Sincronização entre o sinal recebido e a sub-portadora à saída do NCO no sinal transmitido a 1.5 MHz. A vermelho, sub-portadora gerada pelo NCO. A azul, sinal recebido. . . . .	76
5.11	Sincronização entre o sinal recebido e a sub-portadora à saída do NCO no sinal transmitido a 2 MHz. A vermelho, sub-portadora gerada pelo NCO. A azul, sinal recebido. . . . .	76
5.12	Integração do preâmbulo do sinal a 1.5 MHz desmodulado. São visíveis os três integradores em que o integrador central é o relativo à amostragem. Os pontos correspondem aos instantes de amostragem. . . . .	77
5.13	Diagrama de blocos da montagem para testes através do osciloscópio e sem ruído adicionado. . . . .	77
5.14	Sinal de entrada na FPGA a 2 MHz durante uma transição de fase (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima). . . . .	78
5.15	Sinal de correção do <i>Costas loop</i> à entrada do NCO (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima). . . . .	79
5.16	Sinal desmodulado à saída do <i>Costas loop</i> (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima). . . . .	79
5.17	Sinal amostrado e decodificado de NRZI (em baixo) e o trigger relativo à sua amostragem (em cima). . . . .	80

5.18	Diagrama de olho do sinal desmodulado (em baixo) e análise do jitter do trigger relativo à sua amostragem (em cima). . . . .	80
5.19	Diagrama de blocos da montagem para testes através do osciloscópio e com ruído adicionado. . . . .	81
5.20	Sub-portadora gerada pelo <i>Costas loop</i> (em baixo) de um sinal de entrada com SNR de 18 dB (em cima). . . . .	82
5.21	Sinal de correção do <i>Costas loop</i> à entrada do NCO (em baixo) em comparação com um sinal à entrada do sistema com SNR de 18 dB (em cima). . . . .	82
5.22	Sinal integrado do sinal desmodulado (em baixo) confrontado com o sinal amostrado e decodificado de NRZI (em cima) com um sinal à entrada do sistema com SNR de 18 dB. . . . .	83
A.1	Esquema eléctrico da placa implementada de amplificação, filtragem e conversão AD. . . . .	90
A.2	Layout PCB à escala 1:1 da placa implementada de amplificação, filtragem e conversão AD com representação dos componentes. . . . .	91
A.3	Layout PCB à escala 1:1 da placa implementada de amplificação, filtragem e conversão AD. . . . .	92
B.1	Esquema completo da implementação efetuada na FPGA. Todos os blocos (exceto os DCMs) são síncronos pelo relógio de 16 MHz (clk_16mhz) produzido pelo DCM_DV. . . . .	94
C.1	Fotografia da placa de desenvolvimento da Xilinx ML505 com enumeração de secções. . . . .	96
C.2	Fotografia da placa de desenvolvimento da Xilinx ML505 com a placa construída de amplificação, filtragem e conversão analógico-digital. . . . .	97



# Lista de Tabelas

4.1	Ganhos/perdas e balanço de potências em vários pontos da transmissão DSRC	47
4.2	Ganhos/perdas e <i>Noise Figure</i> dos vários componentes do <i>front-end</i> . . . . .	48



# Acrónimos

**ADC** Analog-to-Digital Converter

**AmpOp** Amplificador Operacional

**BER** Bit Error Rate

**BPSK** Binary Phase Shift Keying

**CDR** Clock and Data Recovery

**CEN** Comité Européen de Normalisation

**CLB** Configurable Logic Block

**DAC** Digital-to-Analog Converter

**DCM** Digital Clock Manager

**DPSK** Differential Phase Shift Keying

**DSB-SC** Double-Sideband Suppressed-Carrier

**DSP** Digital Signal Processor

**DSRC** Dedicated Short-Range Communications

**DVB** Digital Video Broadcasting

**EFC** Electronic Fee Collection

**FIR** Finite Impulse Response

**FPGA** Field-Programmable Gate Array

**HDL** Hardware Description Language

**IEEE** Institute of Electrical and Electronics Engineers

**IF** Intermediate Frequency

**IIR** Infinite Impulse Response

**IMD** Intermodulation Distortion  
**ITS** Intelligent Transportation Systems  
**LNA** Low Noise Amplifier  
**LUT** Lookup Table  
**NCO** Numerically-Controlled Oscillator  
**NF** Noise Figure  
**NRZI** Non-Return-to-Zero Inverted  
**OBU** On-Board Unit  
**PLL** Phase-Locked Loop  
**PSD** Power Spectral Density  
**RF** Radio Frequency  
**RSU** Road Side Unit  
**RTTT** Road Transport and Traffic Telematics  
**SAR** Successive-Approximation Register  
**SDR** Software Defined Radio  
**SNR** Signal-to-Noise Ratio  
**VCO** Voltage-Controlled Oscillator  
**VNA** Vector Network Analyzer  
**VSG** Vector Signal Generator

# Capítulo 1

## Introdução

### 1.1 Enquadramento

Nos últimos anos, um dos principais tópicos de preocupação das autoridades tem sido o tráfego e a prevenção rodoviária, de forma a melhorar esta área em termos de custos e eficácia. Com a evolução das tecnologias na área das telecomunicações, têm sido desenvolvidos sistemas que minimizam o tempo perdido em congestionamentos, consumo de energia e danos ambientais, humanos e materiais, contribuindo para a sustentabilidade do setor. Este tipo de sistemas são denominados por *Intelligent Transportation Systems (ITS)*.

Um dos exemplos mais conhecidos é a cobrança eletrónica de taxas, designado internacionalmente por *Electronic Fee Collection (EFC)*. Esta tecnologia permite efetuar o pagamento de taxas de forma automática, sem a necessidade de imobilizar o veículo, tanto na entrada como na saída da infraestrutura. A aplicação mais conhecida desta facilidade de pagamento é na utilização em portagens de autoestradas, pontes, túneis, etc., mas também já é utilizada no pagamento em postos de combustível e em parques de estacionamento.

Para o utilizador, o EFC permite não só evitar o tempo em filas de espera, como também melhorar a sua comodidade. Para o prestador do serviço, tem como principais vantagens evitar a presença de funcionários para este tipo de serviços e a ocupação de menos terreno como consequência da diminuição de corredores de portagens. Todas estes fatores resultam em vantagens monetárias e ambientais.

Este serviço EFC usa comunicações sem fios de curto alcance, normalmente definidas por *Dedicated Short-Range Communications (DSRC)*. Neste caso, as comunicações ocorrem entre os veículos e a infraestrutura.

Esta dissertação visa a implementação de um recetor usado na cobrança eletrónica de portagens, mais especificamente no equipamento da infraestrutura. Com base num equipamento já existente, esta implementação passa pela sua evolução utilizando técnicas digitais na receção de sinais, com recurso ao conceito de *Software Defined Radio (SDR)*. Esta evolução tem como principal objetivo tornar o sistema mais eficaz e com custos mais reduzidos.

## 1.2 Motivação

Recentemente, os *Intelligent Transportation Systems (ITS)* têm se tornado um importante foco de atenção. Estes sistemas terão um importante lugar na segurança e no conforto da condução, estando a ser incentivada a investigação e o desenvolvimento nesta área por todo o mundo. Para o funcionamento destes sistemas são usadas comunicações sem fios DSRC. O DSRC é a base das aplicações ITS como é o caso da cobrança eletrónica de taxas (EFC), informação de viagem, tráfego e segurança, gestão de transportes públicos, e de estacionamento, etc.

Atualmente, existem algumas questões que precisam de ser ultrapassadas para ser possível o desenvolvimento efetivo destes sistemas. Primeiro, é preciso ter em conta que, por estarmos perante sistemas em constante evolução e com forte procura de novas soluções, é necessário que existam mecanismos que flexibilizem os equipamentos usados na troca de informação. Isto permitiria que estes terminais fossem constantemente atualizados sem necessidade de substituição.

Segundo, verificamos que hoje em dia existe a necessidade de adaptação dos dispositivos ao consumidor. Para permitir que os equipamentos se comportem com proximidade às necessidades dos utilizadores, mais uma vez é requerida flexibilidade, sendo assim necessária a existência de mecanismos de auto-adaptação.

Para colmatar este tipo de questões e necessidades, a melhor solução para a evolução dos terminais usados nestas comunicações DSRC é a utilização do conceito de *Software Defined Radio (SDR)*. Isto é, tratar os sinais de rádio tanto quanto possível no domínio digital, e, eventualmente com recurso a *software*. Assim, a motivação para a elaboração desta dissertação passa pela aplicação prática das vantagens do SDR neste tipo de sistemas de rádio, que tipicamente tratavam os sinais apenas com recurso a tecnologias analógicas.

Para além da motivação referida anteriormente, as tecnologias analógicas apresentam algumas desvantagens que podem ser ocultadas com a utilização de SDR. Uma das desvantagens é a presença de ruído introduzido por componentes passivos (resistências, condensadores, etc.) e por componentes ativos (amplificadores) utilizados na implementação destes terminais. Por exemplo nos amplificadores, estes não só introduzem ruído térmico, como também podem provocar distorções não lineares, como é o caso da distorção de intermodulação, *Intermodulation Distortion (IMD)*. De modo geral, esta distorção tem como consequência componentes espectrais indesejáveis derivadas da não linearidade do amplificador, que podem interferir na banda espectral de interesse [Mad01]. Com o recurso ao conceito de SDR, deixam de haver alguns destes problemas derivados do ruído e da não linearidade.

Outra desvantagem dos sistemas analógicos é a complexidade inerente de projetar e implementar filtros, devido a comportamentos não lineares em altas frequências dos componentes eletrónicos que os compõem. Com o uso de SDR, recorrendo a técnicas digitais de construção de filtros, teremos sistemas que eventualmente apresentam melhor cumprimento de máscaras espectrais.

A principal vantagem do SDR é o facto dos dispositivos digitais programáveis terem maior flexibilidade face aos componentes eletrónicos usados na tecnologia analógica. Como resultado, será possível construir sistema atualizáveis, mais adaptáveis e com maior facilidade em realizar testes.

## 1.3 Objetivos

Os objetivos determinados para a elaboração desta dissertação são:

- Estudar os conceitos de comunicações DSRC e *Software Defined Radio (SDR)*;
- Estudar a modulação *Binary Phase Shift Keying (BPSK)* e as várias técnicas para a sua desmodulação;
- Especificar a arquitetura a usar no recetor;
- Simular o projeto;
- Implementar o recetor BPSK em *Field-Programmable Gate Array (FPGA)*;
- Elaborar testes de avaliação e validação para observação e recolha de resultados;

## 1.4 Estrutura da Dissertação

Além do presente capítulo introdutório, esta dissertação está organizada em mais cinco capítulos:

**Capítulo 2 - Conceitos Fundamentais** fornece uma abordagem aos *standards* que envolvem comunicações DSRC a nível europeu. É descrito o funcionamento destas comunicações, tal como as vantagens da aplicação de SDR neste projeto. Também são apresentados conceitos chave sobre componentes a usar.

**Capítulo 3 - Modelação e Simulação do Recetor** são estudadas várias técnicas de desmodulação possíveis para a receção de sinais BPSK. Apresenta também a simulação do recetor a implementar.

**Capítulo 4 - Implementação do Recetor** descreve a forma de implementação do recetor com recurso às técnicas de desmodulação estudadas anteriormente e a componentes frequentes em SDR, nomeadamente *Analog-to-Digital Converter (ADC)* (*hardware* para conversão analógico para digital) e FPGA (dispositivo usado para o processamento digital de sinal).

**Capítulo 5 - Resultados** apresenta os resultados retirados do projeto implementado e descreve o processo usado para os obter.

**Capítulo 6 - Conclusões e Trabalho Futuro** revê a dissertação, conclui o trabalho tendo em conta os objetivos traçados inicialmente, e recomenda trabalho futuro com vista a melhorar a implementação desenvolvida.





## Capítulo 2

# Conceitos Fundamentais

### 2.1 Comunicações DSRC

*Dedicated Short-Range Communications (DSRC)*, em português, comunicações específicas de curto alcance, são comunicações usadas nos *Intelligent Transportation Systems (ITS)*. Temos como exemplo, a cobrança eletrônica de taxas, *Electronic Fee Collection (EFC)*, que usa comunicações entre um veículo e uma infraestrutura.

Os ITS permitiram ainda o despertar de investigação científica de um novo conceito de comunicações – comunicações entre veículos. Este tipo de comunicações com recurso a DSRC tem o objetivo dos veículos partilharem automaticamente informações úteis para o tráfego automóvel, como avisos sobre segurança e controle do trânsito. Estas comunicações utilizam um novo protocolo da família IEEE 802.11 (802.11p), que neste momento se encontra em desenvolvimento, assim como outras soluções proprietárias já desenvolvidas.

Recentemente na cobrança eletrônica de taxas, o tipo de comunicação que este trabalho visa estudar, foram elaboradas formas de coexistirem compatibilidades entre os sistemas EFC já existentes nos vários países da Europa.

Assim, o *Comité Européen de Normalisation (CEN)* criou um grupo técnico de trabalho (TC278) com o objetivo de criar normas que uniformizem a tecnologia EFC nos países europeus.

Das normas realizadas, temos como mais importantes:

- EN 12253 – Define as características ao nível físico a que devem obedecer os sinais DSRC na banda ISM de 5.8GHz;
- EN 12795 – Define as características e mecanismos de acesso ao meio de comunicação (MAC) e controlo lógico da ligação (LLC);
- EN 12834 [ISO 15628] – Define a camada de aplicação e respetivo interface para DSRC;
- EN 13372 – Define os perfis para comunicações DSRC;

Destas, a mais importante a analisar nesta dissertação, é a norma EN 12253, dado que se foca nos requisitos e definições da camada física (PHY), essenciais para a implementação do recetor.

A norma EN 12253:

- especifica a camada física a 5.8 GHz para DSRC aplicável no domínio *Road Transport and Traffic Telematics (RTTT)*, telemática do transporte e tráfego rodoviário.
- fornece especificações sobre a comunicação a ser usada na troca de informação entre as unidades *Road Side Unit (RSU)* e unidades *On-Board Unit (OBU)*.

### 2.1.1 Introdução à Norma EN 12253

As comunicações DSRC/RTTT podem ser consideradas como comunicações *Master-Slave*, onde o *Master* é o RSU e o *Slave* é o OBU. O RSU é o equipamento montado na infraestrutura, isto é, na estrada, e o OBU é o equipamento montado no veículo, que o identifica. A Figura 2.1 exemplifica o caso de uma portagem eletrónica, onde é ilustrada a localização dos equipamentos.

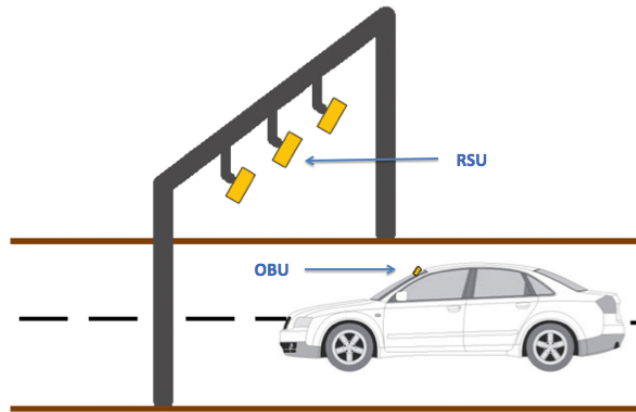


Figura 2.1: Esquema de um sistema EFC.

Existem dois sentidos de comunicação: as comunicações no sentido do RSU para o OBU que são chamadas *Downlink* e as comunicações no sentido do OBU para o RSU que são chamadas *Uplink*. Estas comunicações não ocorrem em simultâneo (Figuras 2.2 e 2.3).

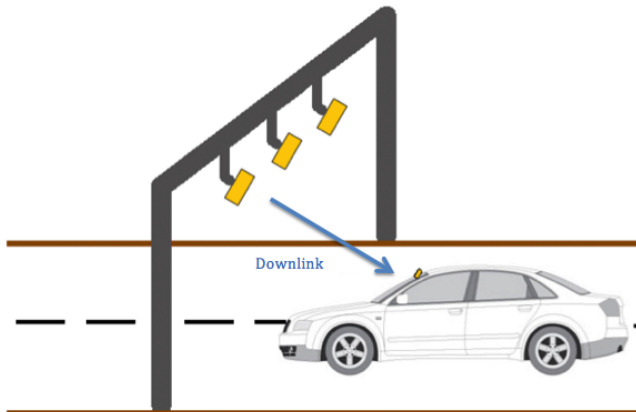


Figura 2.2: Comunicação *Downlink*.

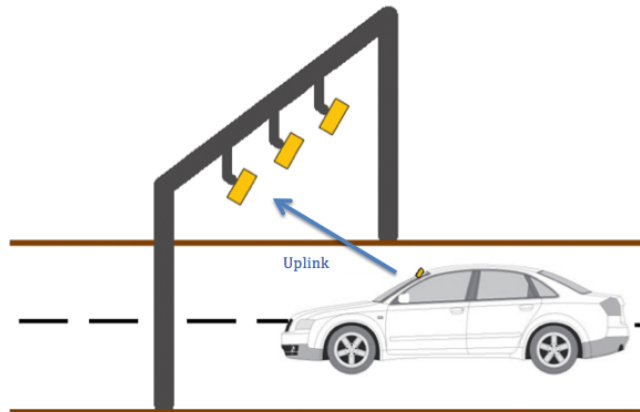


Figura 2.3: Comunicação *Uplink*.

Considerando o enfoque desta dissertação, a implementação de um recetor no RSU, com o objetivo de receber as informações do OBU, que decorre no sentido de transmissão do *Uplink*, nas próximas sub-seções será dada mais importância a este sentido de transmissão.

### 2.1.2 Espectro

Relativamente ao espectro, este tipo de comunicações DSRC operam a 5.8 GHz (banda ISM). Nesta banda, estão disponíveis quatro frequências de portadoras espaçadas entre elas por 5 MHz (Figura 2.4):

- Canal 1: 5.7975 GHz
- Canal 2: 5.8025 GHz
- Canal 3: 5.8075 GHz
- Canal 4: 5.8125 GHz

Destas quatro portadoras, as duas primeiras são reguladas pela norma em questão de forma a serem preferenciais na comunicação, enquanto que as outras duas são opcionais, dependendo do sistema de cada país.

Como é visível na Figura 2.4, a informação transmitida no processo de *Downlink* usa a banda em redor da portadora usada, ou seja usa a portadora diretamente sem qualquer uso de sub-portadoras.

No caso do *Uplink*, a transmissão é modulada através duma sub-portadora de 1.5 MHz ou 2.0 MHz (Figura 2.5). O OBU aproveita a portadora da banda 5.8 GHz gerada pelo RSU no processo de *Downlink*, para modular o sinal a transmitir para *Uplink*, colocando o sinal a 1.5 MHz ou 2.0 MHz centrado nessa portadora. O OBU utiliza uma das opções de frequência da sub-portadora, conforme a instrução que recebe do RSU. Ou seja, esta escolha é dependente do RSU, não estando explícito na norma qual a preferência de sub-portadora a usar.

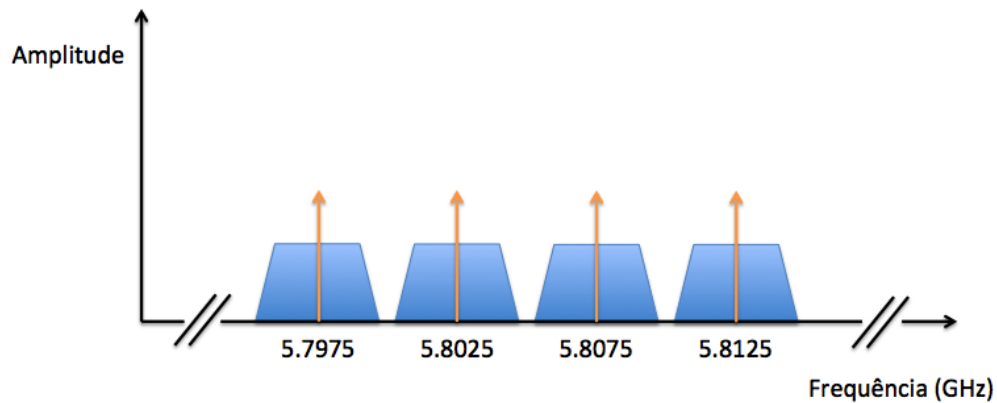


Figura 2.4: Portadoras *Downlink*.

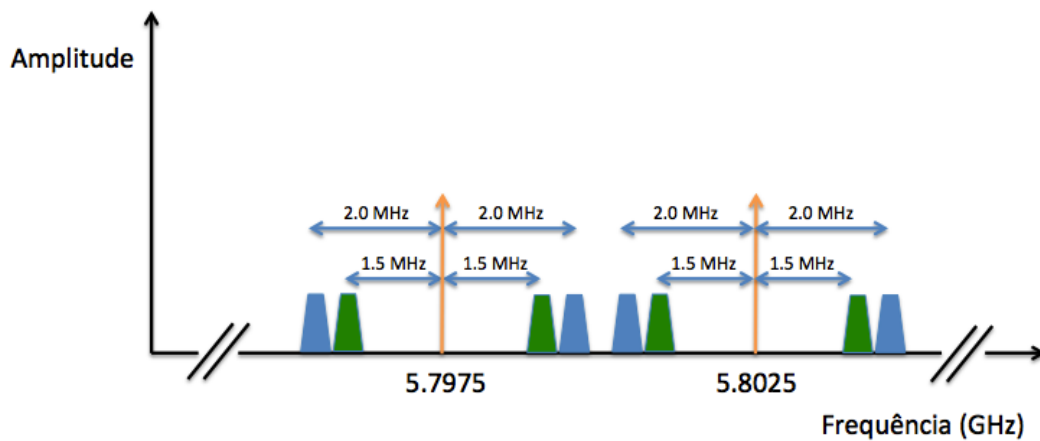


Figura 2.5: Portadoras *Uplink*.

## 2.1.3 Modulação

### 2.1.3.1 *Downlink*

A modulação do *Downlink* é feita em ASK, usando codificação do tipo FM0. A taxa de transmissão em *Downlink* está fixada em 500 kbps.

### 2.1.3.2 *Uplink*

A modulação e codificação em *Uplink* são diferentes do *Downlink*. Com a sub-portadora a 1.5 MHz ou 2.0 MHz, a taxa de transmissão é fixada a 250 kbps.

A modulação usada no *Uplink* é o *Binary Phase Shift Keying (BPSK)*. O tipo de codificação é o *Non-Return-to-Zero Inverted (NRZI)*, que é codificado à transição. Esta codificação funciona da seguinte forma: se no início de um dado período de *bit* houver uma mudança na fase ( $180^\circ$ ) da sub-portadora, então foi transmitido um *bit* '0'; se não ocorrer mudança na fase, foi transmitido um *bit* '1'. As mudanças de fase devem ocorrer na passagem por zero da sub-portadora (Figura 2.6).

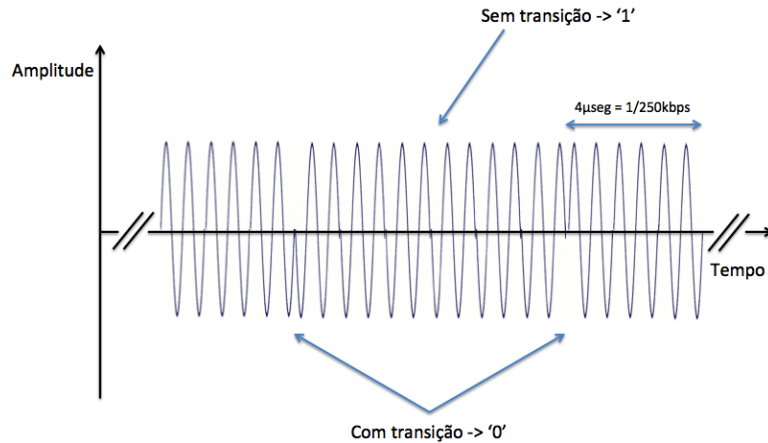


Figura 2.6: Exemplo de aplicação de NRZI na codificação do *Uplink*. Sub-portadora a 1.5MHz.

A recuperação de informação no *Uplink* deverá ser tal que conduza a uma desmodulação com eficácia de recuperação com taxa de erros máxima de  $10^{-6}$ .

#### 2.1.4 Preâmbulo do *Uplink*

O preâmbulo tem como finalidade permitir, ao equipamento que recebe o sinal, a detecção do início de uma nova transmissão e facilitar o ajuste e sincronização do equipamento à mensagem.

Assim, cada trama de informação trocada entre o OBU e o RSU no processo de *Uplink* deverá conter obrigatoriamente um preâmbulo no começo da mensagem.

O preâmbulo do *Uplink* contém duas fases (Figura 2.7):

- 1ª fase consiste em 32 a 36  $\mu s$  de sub-portadora pura, ou seja, sem transições de fase;
- 2ª fase corresponde à transmissão de 8 símbolos '0' BPSK em NRZI, ou seja, oito transições de fase de 180°.

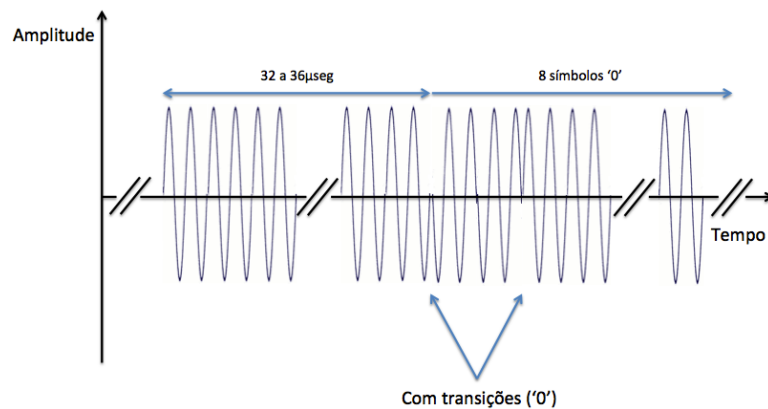


Figura 2.7: Preâmbulo do *Uplink*.

## 2.2 *Software Defined Radio*

Com a rápida evolução tecnológica na área de semicondutores, nas recentes décadas, tornou-se cada vez mais possível a implementação de rádios com recurso a *Software Defined Radio (SDR)*. O SDR foi inicialmente introduzido por Joseph Mitola em 1991, e era descrito da seguinte forma:

*“A software radio is a radio whose channel modulation waveforms are defined in software. That is, waveforms are generated as sampled digital signals, converted from digital to analog via a wideband Digital-to-Analog Converter (DAC) and then possibly upconverted from IF to RF. The receiver, similarly, employs a wideband ADC that captures all of the channels of the software radio node. The receiver then extracts, downconverts and demodulates the channel waveform using software on a general purpose processor.”* [Mit95]

Como descrito pela anterior citação, o SDR é um rádio que usa tecnologia digital tanto quanto possível. Componentes típicos que normalmente eram usados em *hardware* analógico, como amplificadores, filtros e muitos outros, são assim implementados via tecnologia digital, através de, por exemplo, um *Digital Signal Processor (DSP)* ou de uma *Field-Programmable Gate Array (FPGA)*. Este sistema é também caracterizado pela sua flexibilidade e adaptação, devido à facilidade de atualização e reconfiguração do seu comportamento.

Um típico transmissor ou recetor de rádio usa circuitos analógicos dedicados para filtragem, sincronização e modulação / desmodulação. Ora, hoje em dia, há cada vez mais o desejo de conseguir que um só rádio consiga transmitir / receber mais do que um tipo de modulação, como é o caso dos telemóveis, que contêm Bluetooth, Wi-fi, GSM, UMTS, Rádio FM, Infra-vermelhos, etc. Se fossem utilizados apenas circuitos analógicos na construção de rádios com esta complexidade, tornava-se o sistema em algo impraticável, em termos de tamanho, consumo de energia, sistema de refrigeração, etc. É aqui que o SDR entra como solução, pois tal como o conceito indica, este poderá suportar vários tipos de *standards* com recurso a *software* adaptável a vários tipos de modulação.

O sistema “ideal” de SDR é representado pela Figura 2.8. Este sistema representa o objetivo a cumprir, que passa pela adoção de apenas três etapas no projeto de um rádio: antena emissora / recetora, andar de conversão (ADC / DAC) e por fim o sistema digital capaz de amplificar, multiplicar, filtrar, etc. Este sistema “ideal” seria capaz de desmodular qualquer sinal a qualquer frequência de portadora, a qualquer largura de banda e a qualquer tipo de modulação. Representaria assim três características principais: flexibilidade de portadora, flexibilidade de largura de banda e flexibilidade de modulação. No entanto, o estado de arte dos nossos dias ainda não permite esta abstração de tão alto nível de composição de um rádio. Assim, existem propostas para conseguir contornar este problema, com o objetivo de o conseguir o mais “ideal” possível.

Visto o SDR ser o futuro das telecomunicações sem fios, nos anos mais recentes, temos assistido a uma explosão da evolução deste conceito, como por exemplo, no desenvolvimento para aplicações de comunicação e radiodifusão, como são os casos da rede de telemóveis depois da segunda geração e da televisão digital, *Digital Video Broadcasting (DVB)*. Também foi constituído um comité por parte do *Institute of Electrical and Electronics Engineers (IEEE)* com vista ao estudo da criação de *standards* e regras sobre as especificações do SDR devido ao uso do espectro de ondas de rádio público, designado por *IEEE DYSpan Standards Committee* [IEE11].

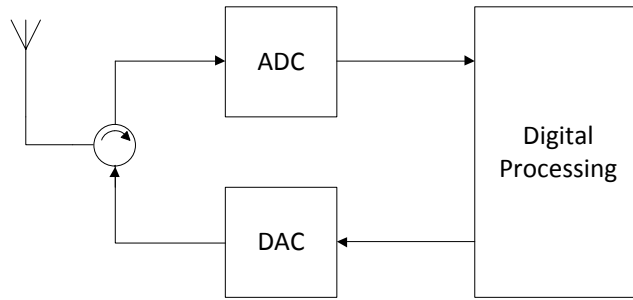


Figura 2.8: A arquitetura SDR ideal.

### 2.2.1 Configurações em SDR

Um dos andares mais importantes num sistema SDR é o processo de digitalização, isto é a conversão de um sinal analógico para digital. Este processo é feito em concordância com o teorema da amostragem de Nyquist-Shannon:

Se uma dada função não contiver frequências superiores a  $B$  Hz, esta é completamente determinada dando as suas ordenadas em séries de pontos espaçados por  $1/(2B)$  segundos. [Man07]

Esta frequência máxima de amostragem  $F_N = 2B$  é designada por frequência de Nyquist.

Idealmente, a digitalização teria lugar logo após a antena, de modo a processar todo o sinal via *software*. No entanto, esta avançada configuração ainda é impraticável, devido a algumas limitações, tal como a reduzida frequência máxima de operação dos dispositivos quando comparada com as elevadas frequências de transmissão.

Assim, como a digitalização à frequência rádio, *Radio Frequency (RF)* é de difícil realização, restam duas propostas. Digitalização em banda base, e digitalização na banda intermédia, *Intermediate Frequency (IF)*.

### 2.2.2 Digitalização em banda base

Digitalização em banda base é a proposta mais recorrente e barata. A configuração é ilustrada através da Figura 2.9.

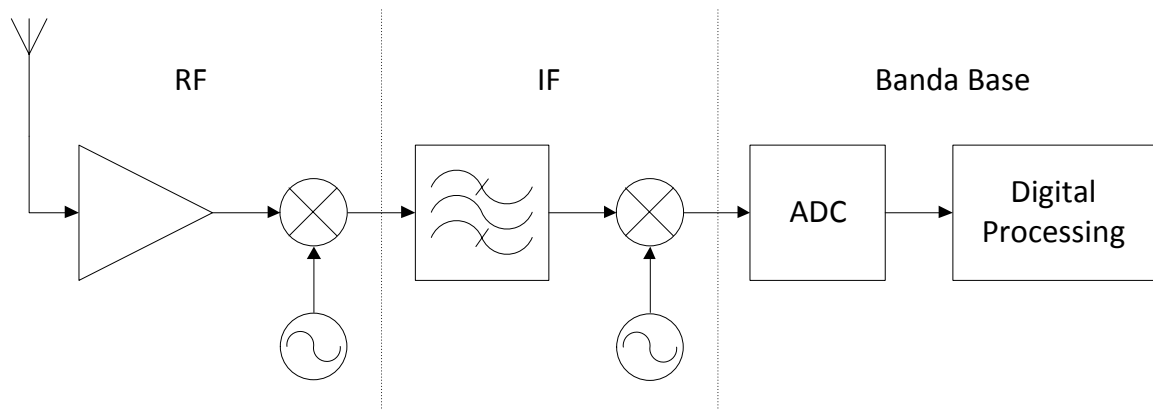


Figura 2.9: Arquitetura simplificada de digitalização na banda base.

O *front-end* utilizado converte o sinal desde a banda RF, a passar pela banda IF até à banda base. Assim, o *front-end* terá que ter pelo menos um andar por cada transição de frequência. Depois de o sinal estar em banda base, este é amostrado pela ADC e as funções, por exemplo, de recuperação de relógio de transmissão e de decodificação do sinal, são implementadas através de tecnologia digital pré-programada.

Pensando no sistema como emissor, em vez de recetor, de forma homóloga, o sistema usará a tecnologia digital disponível em banda base para produzir a codificação do sinal, multiplexagem de canais e modulação do sinal. Depois, o *front-end* tratará de efetuar o *upconvert*, isto é, passar o sinal de banda base para RF, passando pela transição intermédia IF.

### 2.2.3 Digitalização em IF

Devido aos inúmeros progressos da tecnologia semicondutora, é atualmente possível “deslocar mais um pouco” o SDR em direção à antena. Assim, a aposta atual está em fazer uso desta configuração. A configuração é ilustrada através da Figura 2.10.

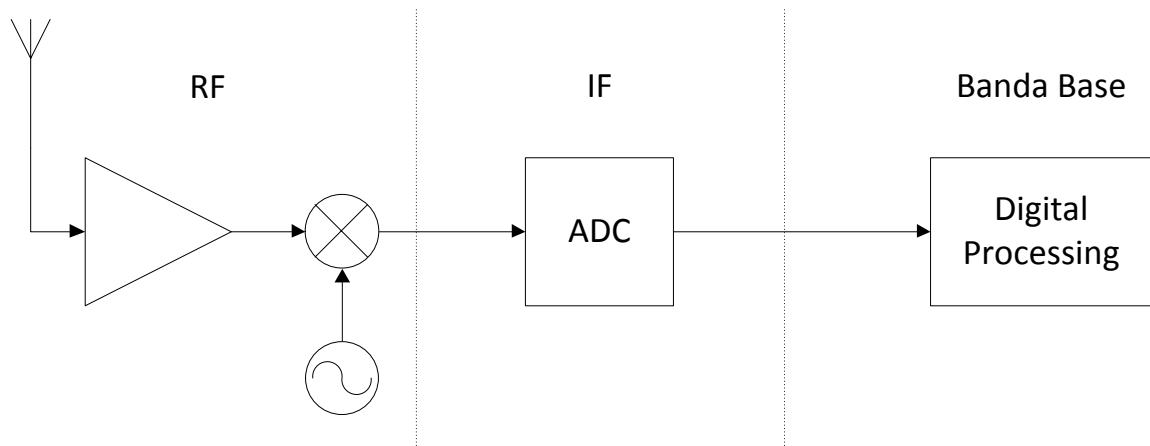


Figura 2.10: Arquitetura simplificada de digitalização em IF.

Nesta configuração o *front-end* irá converter inicialmente o sinal da banda RF para a banda que é processada pela ADC, a banda IF. Em arquitetura simplificada, podemos considerar que é composto principalmente por um amplificador (*Low Noise Amplifier (LNA)*), um oscilador local e um multiplicador.

Posto isto, a ADC amostrará o sinal, e todas as outras funções de rádio serão implementadas via *software*. Estas funções englobam, não só as já anteriormente mencionadas pela configuração anterior, mas também o processo de desmodulação e conversão para banda base, com recurso a osciladores locais e filtros, ambos digitais. Os dispositivos mais usados para a implementação destas funções são DSPs e FPGAs.

Com este acréscimo de funções a serem implementadas via *software*, é possível tornar o sistema mais flexível e ajustável a diferentes frequências e tipos de modulação, tornando esta configuração muito mais desejada.



## 2.2.4 Aplicação de SDR em DSRC

Como analisado durante esta secção, o SDR é caracterizado por ser uma tecnologia rádio onde é possível substituir parte do sistema, que era composto essencialmente por *hardware* analógico, por dispositivos que permitem as comunicações serem tratadas tanto quanto possível por tecnologia digitais.

Posto isto, o SDR é uma solução desejável para o desenvolvimento das comunicações DSRC com vista à melhoria da flexibilidade e adaptação, introduzindo a facilidade de atualização e reconfiguração do comportamento do sistema.

Na próxima secção serão apresentados alguns dos componentes fundamentais a serem usados na implementação de um sistema DSRC com recurso à tecnologia rádio SDR.

## 2.3 Componentes usados em SDR

Embora o conceito de SDR não seja novo, existem diversos fatores que motivaram o enorme desenvolvimento no passar dos últimos anos. Desses fatores, o mais importante é sem dúvida, o contínuo e rápido avanço dos processos e sistemas semicondutores. O resultado final é a disponibilidade de componentes importantes que possibilitam a implementação de rádios com tecnologia SDR, isto é, que fornecem a flexibilidade, robustez, e soluções baratas quando comparados com rádios analógicos convencionais.

### 2.3.1 Analog-to-Digital Converters (ADCs)

A maior parte das aplicações com recurso a ADCs hoje em dia podem ser classificadas em quatro segmentos de mercado: aquisição de dados, medição industrial, voz e áudio, e alta velocidade (geralmente superior a 5 MSPS). Estas aplicações têm como solução três tipos de arquiteturas: *Successive-Approximation Register (SAR)*, *Sigma-delta ( $\Sigma - \Delta$ )*, e *pipelined* ADCs.

A classificação da Figura 2.11 representa de forma geral como estes segmentos de aplicações se relacionam com a resolução da ADC (eixo das ordenadas) e a taxa de amostragem (eixo das abcissas).

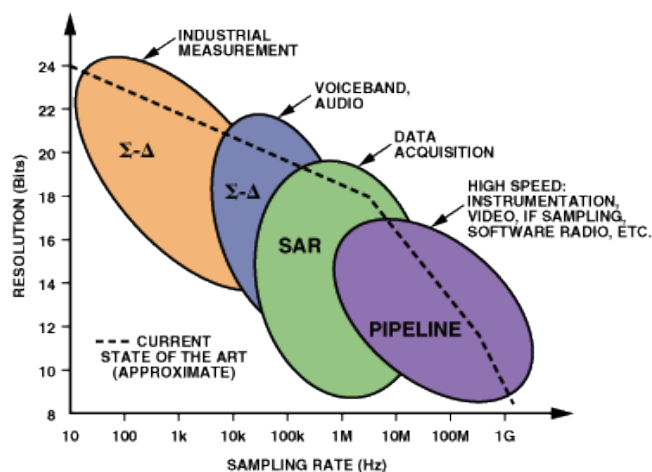


Figura 2.11: Arquiteturas de ADCs, aplicações, resolução e taxas de amostragem. [Kes05]

### 2.3.1.1 Pipelined ADCs

Atualmente, são requeridas ADCs de “alta velocidade” em vários tipos de aplicações de instrumentação (osciloscópios digitais, analisadores de espectro e aparelhos de observação e diagnóstico médico). Também são requeridos conversores de alta velocidade em comunicações (amostragem em IF, SDR, etc.) e eletrônica de consumo (câmeras digitais e televisões de alta-definição, por exemplo).

A ADC *pipelined* teve origem na arquitetura *subranging*, primeiro usada nos anos 1950s. Mais tarde, de forma a aumentar a velocidade de operação da ADC, *subranging* foi desenvolvida a arquitetura *pipelined* ilustrada na Figura 2.12.

Esta ADC *pipelined* tem uma arquitetura *subranging* corrigida digitalmente - em que cada um dos dois estágios opera sobre os dados para metade do ciclo de conversão, e de seguida passa a sua saída residual para a próxima etapa. O *track-and-hold* (T/H) tem como finalidade uma linha de atraso analógica - é programado para entrar no modo de espera quando a conversão da primeira etapa está concluída. Assim permite mais tempo de estabilização para as sub-ADCs, sub-DACs e amplificadores, e permite ao conversor *pipelined* funcionar com uma taxa de amostragem muito maior do que a versão não *pipelined*.

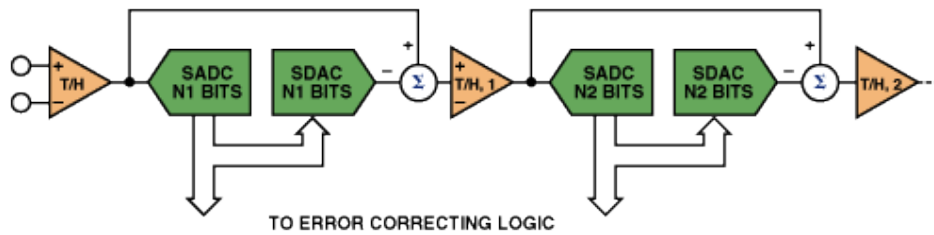


Figura 2.12: Etapas gerais ADCs *pipelined*. [Kes05]

Com vista a garantir que os dados digitais adquiridos individualmente de cada uma das etapas correspondentes a uma amostra particular chegam ao mecanismo de correção em simultâneo, deve ser adicionado um número adequado de *shift registers* a cada uma das saídas das etapas em *pipeline*. Isto adiciona um atraso digital à saída de dados, como é ilustrado pela Figura 2.13, onde se observa o diagrama temporal de uma típica ADC *pipelined* (AD9235 da Analog Devices).

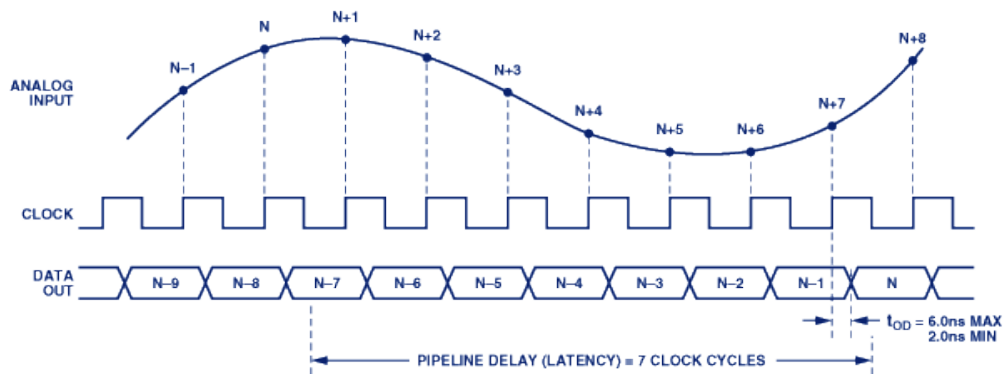


Figura 2.13: Diagrama temporal de uma ADC *pipelined*, de 12 bits, 65 MSPS, a AD9235. [Kes05]

As ADCs *Pipelined* estão hoje em dia disponíveis com resoluções até 16 *bits*, e taxas de amostragem na ordem das centenas de MHz. São ideais para muitas aplicações que requerem não só altas taxas de amostragem, como também um bom *Signal-to-Noise Ratio (SNR)*. A aplicação mais frequente destes conversores é em SDR.

### 2.3.2 *Field Programmable Gate Arrays (FPGAs)*

A FPGA foi introduzida em meados dos anos 1980s como um dispositivo com o objetivo de implementar lógica digital. As FPGAs são otimizadas para circuitos de múltiplos níveis e contêm características especiais de personalização. Estas propriedades permitem adaptação de circuitos muito complexos com recurso a um único *chip*.

#### 2.3.2.1 Arquitetura de FPGAs

Uma FPGA contém uma matriz de elementos lógicos (células) que podem ser interconectados em qualquer configuração desejada para implementar um algoritmo específico.

Por exemplo, nas arquiteturas de FPGAs da família Xilinx, as células lógicas são embutidas como uma “grelha” num *chip*. Este *array* de células é configurado especialmente para existirem ligações rápidas. Esta estrutura é recomendada para sistemas de lógica complexa e irregular, contendo grandes quantidades de possibilidades de ligação.

A Figura 2.14 ilustra um esquema onde é possível visualizar os *Configurable Logic Blocks (CLBs)* existentes, rodeados por outras células como blocos de memória RAM, multiplicadores e blocos de entradas e saídas. Os CLBs são células programáveis implementadas por *Lookup Tables (LUTs)*, *flip-flops* e múltiplos *multiplexers*. As LUTs permitem a criação de funções de saída de combinações arbitrárias das suas entradas.

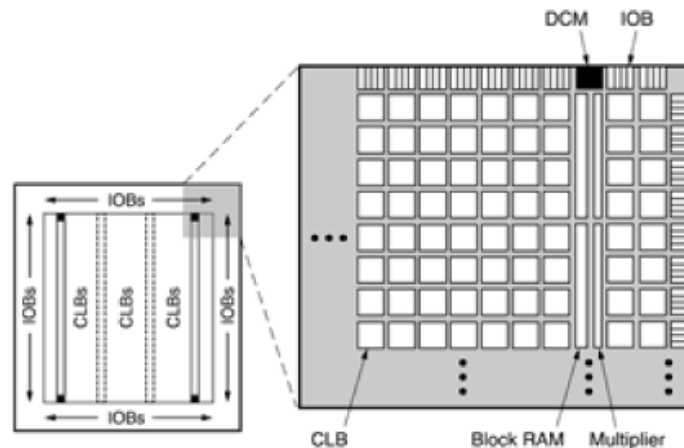


Figura 2.14: Exemplo da estrutura de FPGAs da Xilinx.

#### 2.3.2.2 Comparação entre FPGAs e *Digital Signal Processor (DSPs)*

As tecnologias digitais de SDR podem ser implementadas também através de DSPs. O DSP é essencialmente um microprocessador otimizado para a aplicação em sistemas de processamento digital de sinal. Os DSPs são bastante flexíveis porque podem ser programados de forma fácil e rápida com recurso a linguagens de alto nível, como por exemplo C.

No entanto, trazem algumas desvantagens em relação às FPGAs. Por exemplo, nas aplicações que requerem elevado número de operações em paralelo, as operações demoram mais tempo já que o suporte a operações paralelas é muito reduzido. Neste caso, pelas FPGAs serem mais flexíveis no nível mais baixo de implementação, é possível projetar o sistema com o número de recursos paralelos pretendidos.

### 2.3.2.3 Aplicação de FPGAs em SDR

Em SDR, as FPGAs têm a principal vantagem de conseguir adicionar flexibilidade ao produto final. As FPGAs por vezes até podem contribuir para a diminuição da área de silício a usar, pois com um único *chip* consegue-se realizar todo o algoritmo do sistema. As situações em que é benéfico o uso de FPGAs em processamento digital de sinal são apresentadas na seguinte lista [Ree02]:

- Sistemas com elevadas taxas de amostragem;
- Sistemas que envolvam tamanho de palavras variáveis ou pouco convencionais, pois com as FPGAs torna-se bastante eficiente, já que é definido para cada sinal o tamanho de *bits* pretendido;
- Sistemas com filtros *Finite Impulse Response (FIR)* de elevada ordem, pois o algoritmo pode ser implementado em paralelo, diminuindo o tempo de execução.

### 2.3.2.4 Implementação de Projetos em FPGAs

São geralmente necessários alguns passos no processo de implementação em FPGA. De seguida, são enumerados os processos típicos que compõem o fluxo de projeto em FPGAs (com o auxílio da Figura 2.15):

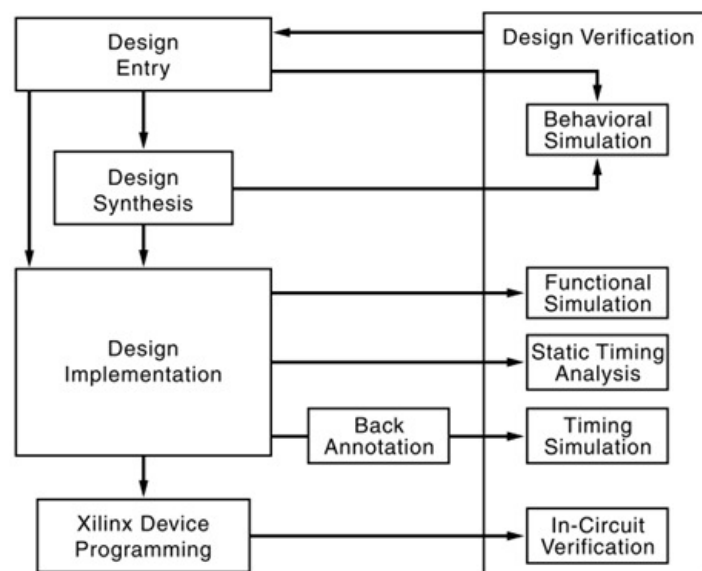


Figura 2.15: Fluxo de projeto em FPGAs.

1. Modelar o circuito com recurso a, por exemplo, programação em *Hardware Description Language (HDL) (Design Entry)*.
2. Sintetizar. Sintetiza linguagens VHDL, Verilog ou outro tipo de linguagens de projeto para criar um ficheiro *netlist (Design Synthesis)*.
3. Verificar a funcionalidade dos modelos criados através de simulação (*Behavioral Simulation*).
4. Implementar. Traduz todos os modelos criados através de programação HDL, ou até, de modelos criados através de *Cores* (ferramentas disponibilizadas pelo fabricante). Mapeia e faz o *routing* dos diversos blocos (*Design Implementation*).
5. Gerar os ficheiros em código máquina. Estes ficheiros serão transferidos para a FPGA com o objetivo de configurar os seus elementos (*Xilinx Device Programming*).
6. Testar a execução do projeto (*In-Circuit Verification*).

Diversas ferramentas são usadas para desempenhar estas funções. No caso da Xilinx, a ferramenta mais usada é o ISE, que incorpora grande parte destas funcionalidades.

### 2.3.2.5 Filtros *Finite Impulse Response (FIR)* em FPGAs

O filtro FIR é um tipo de filtro de processamento de sinal cuja resposta impulsional é de duração finita, pois torna-se nulo ao fim de um determinado tempo. É assim diferente dos filtros *Infinite Impulse Response (IIR)*, que têm malha fechada interna e que continuam a responder indefinidamente (geralmente a decair). A resposta impulsional de um filtro FIR de tempo discreto e de ordem  $N$  tem uma duração de  $N+1$  amostras.

Um filtro FIR tem algumas vantagens que geralmente torna-o preferível ao filtro IIR [Man07]:

- Não requer malha de realimentação. Isto significa que erros de arredondamento não se propagam pelos ciclos de malha-fechada não interferindo nos cálculos seguintes. Isto torna também a implementação bastante mais simples.
- São inerentemente estáveis. Isto é também devido ao facto de não haver malhas de realimentação. Forma-se uma única linha de execução.
- Podem ser facilmente projetados para serem lineares em fase, colocando os coeficientes simétricos. Isto corresponde ao facto da fase ser proporcional à frequência, correspondendo a um atraso igual a todas as frequências. Esta propriedade é muito útil para certas aplicações com sensibilidade de fase, como é o caso das comunicações.

A versão convencional de um filtro FIR *single-rate* é definida através do somatório da equação 2.1, onde  $N$  é o número de coeficientes do filtro:

$$y(k) = \sum_{n=0}^{N-1} a(n)x(k-n) \quad k = 0, 1, \dots \quad (2.1)$$

Na família de FPGAs da Xilinx, este tipo de filtros pode ser implementado através de *Cores* destinado para este fim. A Figura 2.16 ilustra a linha de atraso convencional do filtro FIR. No entanto, se esta ilustração é uma concepção útil no cálculo realizado pelo *Core*, a realização em FPGA é um pouco diferente.

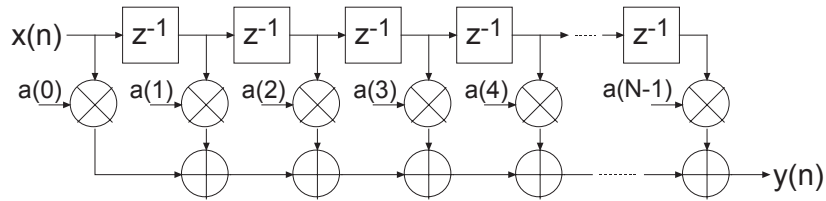


Figura 2.16: Representação do atraso convencional da linha do filtro FIR [Xil10b]

A Figura 2.17 ilustra o diagrama temporal de um filtro *single-rate*, de um canal. É possível verificar o número de ciclos de atraso entre a entrada (DIN) de uma amostra e a saída (DOUT) correspondente.

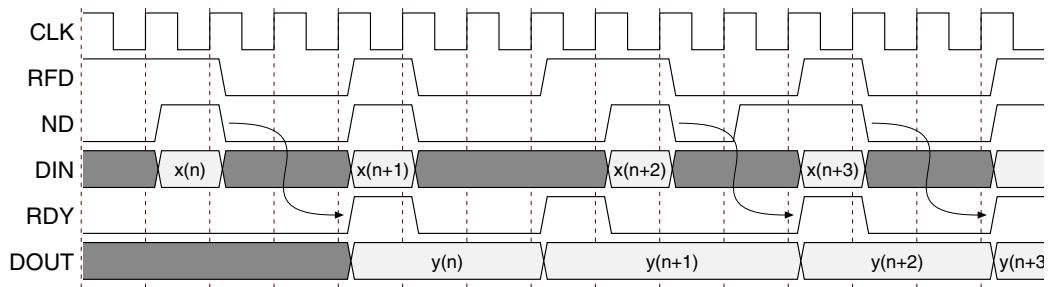


Figura 2.17: Diagrama temporal de um filtro FIR em FPGA. [Xil10b]

## Capítulo 3

# Modelação e Simulação do Recetor

### 3.1 Introdução

Como apresentado anteriormente, a modulação usada na transmissão de sinais do *Uplink* do sistema DSRC é o BPSK.

Durante este capítulo serão estudadas várias técnicas de desmodulação para a receção de sinais BPSK que depois serão aplicadas em simulações através do MATLAB.

Por fim, através da comparação entre as várias técnicas, será escolhida qual destas será utilizada na implementação do recetor.

### 3.2 O BPSK

Dada uma portadora sinusoidal,

$$A_p \cos(2\pi f_p t + \theta) \quad (3.1)$$

A informação pode ser transmitida por variação da amplitude  $A_p$ , frequência angular  $\omega_p$  ( $2\pi f_p$ ) ou fase  $\theta$ . Na modulação binária de fase, designada por *Binary Phase Shift Keying* (*BPSK*), a portadora modulada  $x_p(t)$  tem a seguinte forma:

$$x_p(t) = A_p x(t) \cos(\omega_p t) \quad (3.2)$$

Sendo  $x(t)$  o sinal modulante que toma valores do conjunto  $\{-1, 1\}$  e  $\cos(\omega_p t)$  a portadora.

Esta modulação é considerada como transmissão *Double-Sideband Suppressed-Carrier* (*DSB-SC*), em que as frequências produzidas por modulação são simetricamente espaçadas antes e depois da frequência da portadora e o nível da amplitude da portadora é reduzida para o menor nível possível, de preferência completamente suprimida [Hay00].

Assim, a portadora é dada por (quando representa um símbolo '1'):

$$x_p(t) = A_p \cos(\omega_p t) \quad (3.3)$$

E dada por (quando representa um símbolo '0'):

$$x_p(t) = -A_p \cos(\omega_p t) \quad (3.4)$$

A Figura 3.1 apresenta uma forma de onda BPSK.

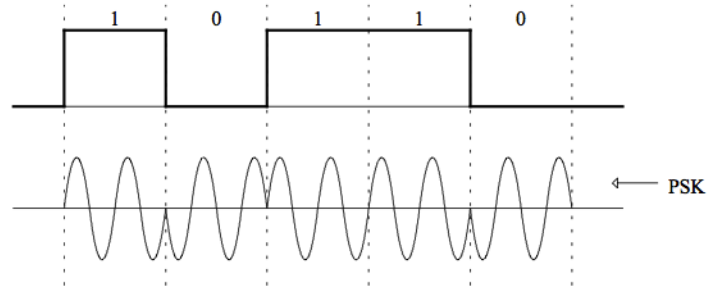


Figura 3.1: Exemplo de forma de onda BPSK. No topo, o sinal em banda base, em baixo o sinal em BPSK.

O sinal BPSK pode ser considerado como resultante da multiplicação do sinal NRZI bipolar,  $p(t)$ , pela portadora  $\cos(\omega_p t)$ .

Assim, a expressão do sinal BPSK será:

$$x_p(t) = A_p p(t) \cos(\omega_p t) \quad (3.5)$$

Com  $p(t)$  o sinal banda-base com codificação NRZI.

A *Power Spectral Density (PSD)*, em português, densidade espectral de potência (DEP), do sinal BPSK terá portanto a forma de banda lateral dupla. A Figura 3.2 mostra a PSD apenas nas frequências positivas.

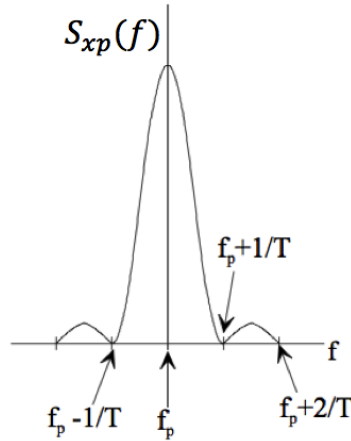


Figura 3.2: PSD nas frequências positivas de um sinal BPSK.

Da Figura 3.2 conclui-se que a largura de banda do sinal BPSK é dada por

$$B_T = 2B \quad (3.6)$$

Onde  $B$  é a largura de banda sinal modulante  $p(t)$ .



### 3.3 Desmodulação BPSK

A receção de sinais BPSK é efetuada por detetores chamados coerentes ou síncronos. Num detetor síncrono multiplica-se o sinal de entrada por uma portadora gerada localmente [Hay00]. É assim representado um esquema deste tipo de detetor na Figura 3.3.

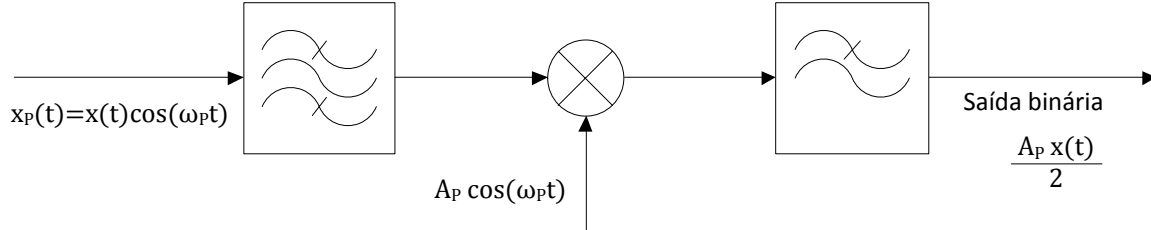


Figura 3.3: Esquema de um detetor síncrono para sinais BPSK.

O inconveniente dos sistemas síncronos é que exigem uma boa sincronização em frequência e em fase do oscilador local com o sinal de entrada. Para mostrar isto, considera-se que a portadora local tem uma frequência  $\omega_p + \Delta\omega$ , isto é, um erro de frequência igual a  $\Delta\omega$ . No caso de um sinal BPSK tem-se:

$$A_p x(t) \cos(\omega_p t) \cos((\omega_p + \Delta\omega)t) = \frac{A_p}{2} x(t) [\cos((2\omega_p + \Delta\omega)t) + \cos(\Delta\omega t)] \quad (3.7)$$

O resultado da expressão 3.7 mostra o resultado matemático à saída do multiplicador do detetor.

Depois do filtro passa-baixo tem-se:

$$\frac{A_p}{2} x(t) \cos(\Delta\omega t) \quad (3.8)$$

Logo, tem-se como resultado um sinal em banda-base transladado na frequência de  $\Delta\omega$ .

Por outro lado se a portadora local tiver um erro de fase  $\theta$ , isto é se ela for definida por  $\cos(\omega t + \theta)$  teremos à saída do filtro passa-baixo do recetor BPSK:

$$\frac{A_p}{2} x(t) \cos(\theta) \quad (3.9)$$

Pela expressão 3.9, é possível concluir que se:

- $\theta = 0$ , então obtém-se o sinal desejado;
- $\theta = \frac{\pi}{2}$ , então obtém-se a saída nula;
- $\theta > \frac{\pi}{2}$ , então  $x(t)$  inverte a sua polaridade pelo que os 1's transformam-se em 0's e os 0's em 1's.

Daqui se conclui que o sincronismo de fase é também muito importante.

De notar que o sincronismo é muito difícil de obter em sistemas modulados quando comparados com sistemas de banda base. Nestes últimos a sincronização é efetuada por dentro de um intervalo de um *bit*; enquanto que nos sistemas modulados a sincronização deverá ser efetuada à escala de  $\frac{1}{f_p} \ll T$ , sendo portanto muito mais difícil de concretizar.

Utilizam-se vários métodos para obter a informação necessária para a sincronização:

1. Uso de um padrão temporal;
2. Uso de um sinal de sincronismo separado;
3. Uso de autossincronização. Este método consiste na extração de informação de portadora diretamente do sinal modulado.

No primeiro método o emissor e recetor são controlados por uma fonte de temporização precisa. Este método é muito utilizado em grandes redes de dados, mas é raramente utilizado em ligações ponto-a-ponto.

Em ligações ponto-a-ponto podem ser utilizados sinais de sincronismo separados na forma de portadoras piloto. Estas são enviadas em conjunto com o sinal modulante utilizando a *multiplexagem* na frequência ou no tempo. Em cada um destes, o sinal de sincronismo é isolado no recetor e utilizado para sincronizar o oscilador local e controlar as operações de amostragem no recetor.

No entanto, perante a norma já analisada, no capítulo anterior, que guiará a implementação deste trabalho, nenhum dos métodos anteriores corresponde às especificações apresentadas.

Assim, é necessário recorrer ao método de autossincronismo. Em autossincronismo, a portadora local de referência e a informação de temporização é extraída a partir das ondas moduladas. Dos tipos de autossincronismo existentes, vão ser analisados os dois seguintes:

- Extração de portadora pelo quadrado do sinal recebido;
- *Costas loop*.

### 3.3.1 Extração de portadora pelo quadrado do sinal recebido

A Figura 3.4 representa o diagrama de blocos de um circuito que obtém uma portadora local coerente a partir de um sinal BPSK. Pela análise das expressões dos sinais à saída de cada um dos blocos constituintes, facilmente se conclui que à saída se obtém um sinal sinusoidal com a mesma frequência da portadora.

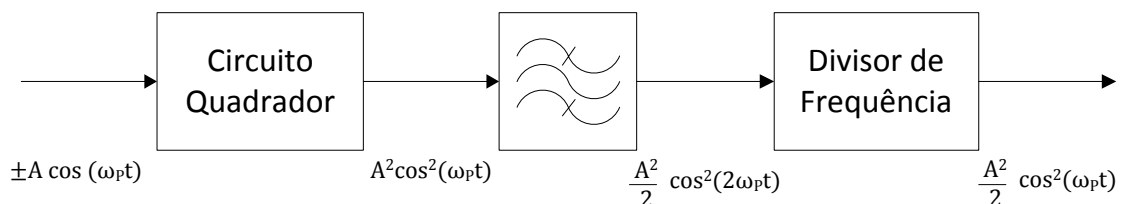


Figura 3.4: Esquema de um recuperador de portadora de sinal BPSK através de um circuito quadrador do sinal de entrada.

Então, um recetor BPSK completo será o indicado na Figura 3.5, onde se inclui o circuito extrator da portadora.

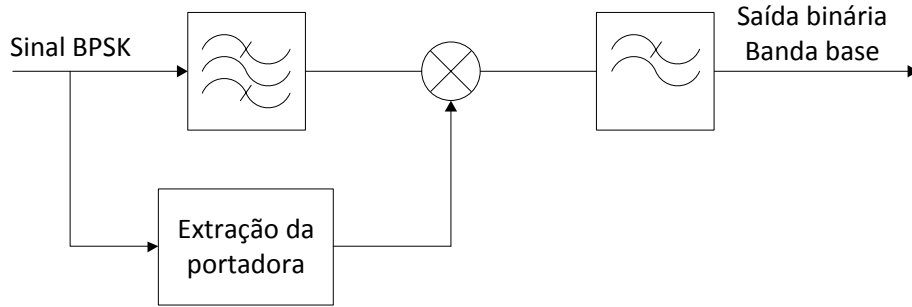


Figura 3.5: Esquema de um recetor BPSK com recuperação de portadora através de um circuito extrator.

O circuito extrator da portadora analisado tem contudo um problema – a fase da portadora extraída tem uma ambiguidade de fase igual a  $\pi$ . De facto, a portadora extraída é sempre  $\frac{A^2}{2} \cos(\omega_p t)$  para as duas situações seguintes:

$$\pm A \cos(\omega_p t) \quad (3.10)$$

$$\pm A \cos(\omega_p t + \pi) \quad (3.11)$$

No caso da situação 3.10, a portadora extraída está em fase com a portadora utilizada no emissor. Já na situação 3.11, a portadora extraída está desfasada de  $\pi$  relativamente à portadora do emissor. Como foi visto anteriormente, a um desfasamento de  $\pi$  entre as duas portadoras corresponde uma inversão do sinal à saída do desmodulador BPSK, pelo que existem erros em todos os *bits* recebidos.

Este problema é resolvido pela utilização da codificação NRZI já referida anteriormente. Supondo que existe erro de fase na portadora do recetor, este não é refletido no fim do sistema. A decodificação em banda-base é efetuada pela ocorrência ou não de transição de fase, não interferindo se o sinal recebido está desfasado em  $\pi$  radianos.

No entanto, esta alternativa de recuperação de portadora, apresenta alguns inconvenientes, tais como:

- Dificuldade de sincronização entre a portadora recuperada e o sinal recebido após o filtro passa-banda. Visto que no domínio digital cada um destes blocos tem algum atraso de amostras, é necessária a sincronização entre os dois sinais referidos. Embora possa ser determinada uma previsão deste atraso por se conhecer previamente o atraso de cada bloco, será explicado em detalhe o porquê da dificuldade no sistema a implementar.
- Necessidade de alguns filtros passa-banda com elevada ordem. Isto pode ser um problema para a implementação, já que é necessária alguma prevenção com os recursos disponíveis na FPGA. O uso de filtros passa-banda de elevada ordem pode provocar uma saturação no número de multiplicadores dedicados existentes na FPGA.
- Não contém realimentação e assim não aproveita a possibilidade de caminhar para uma situação de estabilidade onde é possível atenuar alterações bruscas que podem ocorrer no sistema.

Estes inconvenientes serão analisados em detalhe na simulação, onde se trata da análise das implementações dos dois métodos de recuperação de portadora.

### 3.3.2 Costas loop

Este método de desmodulação baseia-se numa *Phase-Locked Loop (PLL)*<sup>1</sup> e é usado para a recuperação de portadora na receção de transmissões de sinais DSB-SC, como é o caso da transmissão BPSK. Foi inventado por John P. Costas em meados dos anos 1950 [Cos02].

O esquema deste tipo de desmodulador é representado pela Figura 3.6.

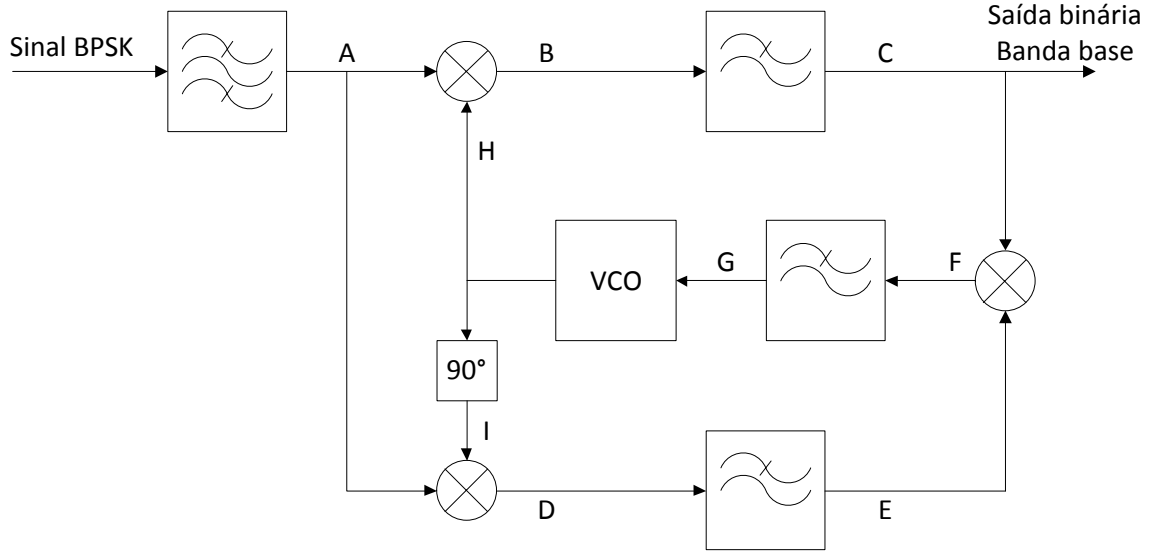


Figura 3.6: Esquema completo de um recetor BPSK pelo método *Costas loop*.

Da análise da Figura 3.6, retiramos que estamos perante um desmodulador bem mais complexo. Embora seja mais complexo de explicar, o funcionamento é bastante simples.

O primeiro facto a ser notado é que estamos essencialmente perante dois recetores básicos com o mesmo sinal de entrada mas com um oscilador local de onde se obtêm dois sinais em quadratura entre si (na Figura 3.6 em H e I):

$$\cos(\omega_p t) \leftarrow H \quad (3.12)$$

$$\cos\left(\omega_p t + \frac{\pi}{2}\right) \leftarrow I \quad (3.13)$$

Para perceber melhor a operação do *Costas loop*, consideramos inicialmente que o oscilador local está em fase em relação ao sinal recebido. Nestas condições, a saída do multiplicador em fase conterà o sinal em banda base desmodulado. Considerando o sinal de entrada depois do filtro passa-banda<sup>2</sup> (A):

$$A_p p(t) \cos(\omega_p t) \leftarrow A \quad (3.14)$$

<sup>1</sup>Phase-locked loop (PLL), em português, Malha de Captura de Fase, é um sistema de realimentação em que o sinal de realimentação é usado para sincronizar a frequência do sinal de saída com a frequência do sinal de entrada.

<sup>2</sup>Este filtro tem o propósito de filtrar ruído fora da banda de interesse, tal como rejeitar frequências indesejadas introduzidas pelo *front-end*, na existência de um antes do desmodulador em questão.

À saída do multiplicador em fase (B):

$$A_p p(t) \cos^2(\omega_p t) \quad (3.15)$$

$$= \frac{A_p p(t)}{2} + \frac{A_p p(t)}{2} \cos(2\omega_p t) \quad \leftarrow B \quad (3.16)$$

E depois de passar pelo filtro passa-baixo, em C, teremos o sinal em banda-base:

$$\frac{A_p p(t)}{2} \quad \leftarrow C \quad (3.17)$$

Enquanto que o resultado à saída do multiplicador em quadratura (D):

$$A_p p(t) \cos(\omega_p t) \cos\left(\omega_p t + \frac{\pi}{2}\right) \quad (3.18)$$

$$= \frac{A_p p(t)}{2} \left( \cos\left(\frac{\pi}{2}\right) + \cos\left(2\omega_p t + \frac{\pi}{2}\right) \right) \quad \leftarrow D \quad (3.19)$$

Que depois do filtro passa-baixo (E), produzirá um sinal nulo:

$$\frac{A_p p(t)}{2} \cos\left(\frac{\pi}{2}\right) = 0 \quad \leftarrow E \quad (3.20)$$

Ora, no cálculo da amplitude do erro, no multiplicador da malha (F), o sinal será nulo, logo o *Voltage-Controlled Oscillator (VCO)*<sup>3</sup> não sofrerá correção:

$$\frac{A_p p(t)}{2} \cdot 0 = 0 \quad \leftarrow F \quad (3.21)$$

Se agora pensarmos num oscilador local desfasado alguns graus ( $\theta$ ), em A, temos:

$$A_p p(t) \cos(\omega_p t + \theta) \quad \leftarrow A \quad (3.22)$$

À saída do multiplicador em fase (B):

$$A_p p(t) \cos(\omega_p t + \theta) \cos(\omega_p t) \quad (3.23)$$

$$= \frac{A_p p(t)}{2} (\cos(\theta) + \cos(2\omega_p t + \theta)) \quad \leftarrow B \quad (3.24)$$

E depois de passar pelo filtro passa-baixo, em C, teremos:

$$\frac{A_p p(t)}{2} \cos(\theta) \quad \leftarrow C \quad (3.25)$$

Com o resultado à saída do multiplicador em quadratura (D):

$$A_p p(t) \cos(\omega_p t + \theta) \cos\left(\omega_p t + \frac{\pi}{2}\right) \quad (3.26)$$

---

<sup>3</sup>Voltage-controlled oscillator (VCO) – é um oscilador eletrónico projetado com o objetivo de controlar a frequência de oscilação pela amplitude da tensão de entrada. No projeto em questão, este componente vai ser substituído por um Numerically-controlled oscillator (NCO), versão digital de um VCO, de modo a poder ser aplicado na FPGA. O NCO será explicado com mais detalhe no capítulo de implementação do recetor.

$$= \frac{A_p p(t)}{2} \left( \cos \left( \theta + \frac{\pi}{2} \right) + \cos \left( 2\omega_p t + \theta + \frac{\pi}{2} \right) \right) \leftarrow D \quad (3.27)$$

E agora à saída do filtro passa-baixo a seguir ao multiplicador em quadratura (E) aparecerá uma pequena amplitude:

$$\frac{A_p p(t)}{2} \cos \left( \theta + \frac{\pi}{2} \right) \leftarrow E \quad (3.28)$$

Ora, no cálculo do erro no multiplicador dentro da malha (F), aparece então uma amplitude, que servirá para corrigir a frequência do VCO de modo a compensar o erro de fase:

$$\frac{A_p^2 p(t)^2}{4} \cos(\theta) \cos \left( \theta + \frac{\pi}{2} \right) \quad (3.29)$$

$p(t)^2 = 1$  visto que  $p(t)$  toma valores do conjunto  $\{-1, 1\}$ .

$$= \frac{A_p^2}{8} \cos \left( \frac{\pi}{2} \right) + \cos \left( 2\theta + \frac{\pi}{2} \right) \quad (3.30)$$

$$= \frac{A_p^2}{8} \cos \left( 2\theta + \frac{\pi}{2} \right) \leftarrow F \quad (3.31)$$

O filtro passa-baixo que vem a seguir serve essencialmente para filtrar altas frequências que inevitavelmente se propagam à saída dos filtros anteriores, por estes não serem ideais. Este filtro torna assim o erro mais estável em G, produzindo uma amplitude DC.

Através da análise da expressão resultante em 3.31, variando  $\theta$ , verifica-se que esta expressão repete-se em  $\pi$ . Considera-se assim o domínio da expressão de  $[-\frac{\pi}{4}, \frac{3\pi}{4}]$ .

Existem assim quatro situações extremas:

- $\theta = -\frac{\pi}{4} \rightarrow G = \frac{A_p^2}{8}$
- $\theta = 0 \rightarrow G = 0$
- $\theta = \frac{\pi}{4} \rightarrow G = -\frac{A_p^2}{8}$
- $\theta = \frac{\pi}{2} \rightarrow G = 0$

A segunda situação já foi analisada anteriormente. Pela análise da primeira e terceira situações, verifica-se que a amplitude à entrada do VCO contraria o erro de fase verificado no sinal recebido. Quando o sinal vem desfasado  $\theta = -\frac{\pi}{4}$ , ou seja, adiantado em relação ao oscilador do VCO, então a amplitude à entrada do VCO será positiva de modo a provocar uma aceleração de frequência até capturar o sinal recebido. Quando o sinal vem desfasado  $\theta = \frac{\pi}{4}$ , a amplitude à entrada do VCO é negativa de modo a provocar uma desaceleração em frequência.

Na última situação, embora o erro dê 0 tal como acontece quando o sinal está em fase com a sinusóide do VCO, admitindo que o erro de fase nunca é precisamente  $\frac{\pi}{2}$ , mas sim  $\frac{\pi}{2}^-$  ou  $\frac{\pi}{2}^+$ , a malha tem tendência a fugir da amplitude de erro de 0. A derivada neste ponto na passagem por 0, é contrária em comparação com quando o erro de fase é 0 ou  $\pi$ . Isso é verificado pela inclinação das curvas que pode ser visível na Figura 3.7.

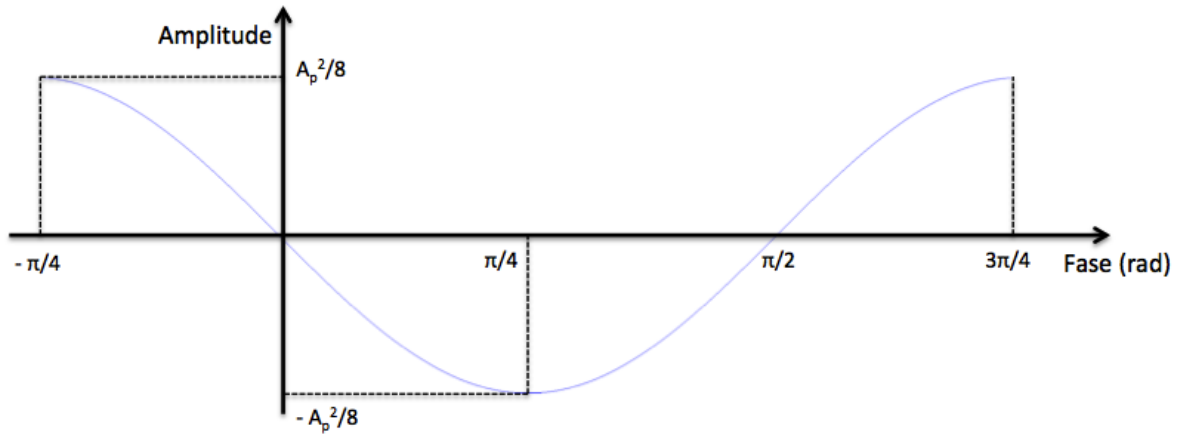


Figura 3.7: Dependência da amplitude à entrada do VCO com o erro de fase do sinal recebido.

É ainda possível verificar, considerando a não existência de ganho por parte dos blocos constituintes do desmodulador (multiplicadores e filtros) à frequência de interesse, que a amplitude à entrada do VCO tem a seguinte relação com a amplitude do sinal recebido:

$$\frac{\frac{A_p^2}{8}}{A_p} = \frac{A_p}{8} \quad (3.32)$$

Deste resultado, pode-se retirar uma vantagem deste método face a outros, que por exemplo, utilizem apenas uma PLL – o sinal de correção à entrada do VCO é independente da fase do sinal modulado em banda-base,  $p(t)$ , pois este “desaparece” no multiplicador à entrada na malha, como verificado nas expressões 3.29 a 3.31. Assim, quando ocorrem mudanças de fase, a correção resultante é imune a tal inversão de  $180^\circ$ , mantendo-se a malha de realimentação estável. No entanto, este resultado denota ainda um problema – a amplitude de erro à entrada do VCO está dependente do conhecimento da amplitude do sinal recebido à entrada do *Costas loop*. Este problema é contudo possível de resolver, e a resolução será detalhada no capítulo da implementação do recetor.

Finalmente, depois da apresentação de uma breve explicação deste método, conclui-se que este é notoriamente mais eficaz e recomendável que o método anterior. As principais razões são o facto do *Costas loop* recorrer à existência de realimentação, corrigindo o erro de forma automática e sistemática, e o facto de usar apenas filtros passa-baixo, possíveis de implementar com um número de coeficientes relativamente baixo.

### 3.4 Recuperação do relógio de transmissão em banda-base

Depois da desmodulação do sinal recebido, assim que este se encontra em banda-base, é necessário amostrar e decodificar os símbolos, já que estes se encontram codificados em NRZI.

Mas, tal como no processo de desmodulação, também no processo de amostragem em banda-base é necessário haver sincronização. Esta sincronização é possível se for projetada uma forma de conhecer o relógio de transmissão à qual o sinal foi produzido em banda-base antes da modulação em BPSK.

Esta sincronização é no entanto mais fácil de obter do que no processo de desmodulação, pois estando o sinal em banda-base, o período  $T$  de cada símbolo é relativamente superior aos tempos com que foram lidados no processo anterior.

A recuperação do relógio de transmissão começa por ter um integrador de tempo finito, em que esse tempo é a duração do período de um símbolo. O esquema deste recuperador e sincronizador do relógio de banda-base é ilustrado na Figura 3.8.

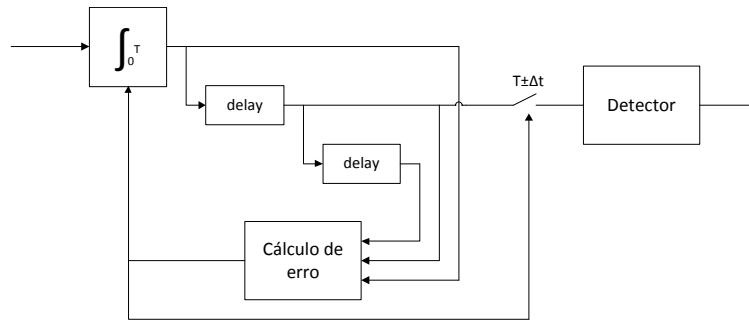


Figura 3.8: Esquema do recuperador de relógio dos símbolos transmitidos em banda-base.

Os blocos da Figura 3.8 são descritos da seguinte forma:

**Integrador de tempo finito** integra o sinal recebido durante um determinado período  $T$ . Funcionando como um filtro, este bloco torna a onda quadrada recebida, numa onda triangular. Isto permite a criação de máximos e mínimos, tornando perceptível ao sistema a deteção do relógio. O esboço do resultado deste integrador é apresentado na Figura 3.9.

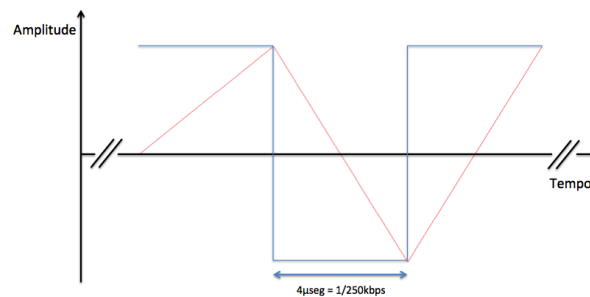


Figura 3.9: Esboço do funcionamento do integrador de tempo finito. A azul, sinal em banda-base recebido à entrada do integrador. A vermelho, sinal integrado em tempo finito  $T = 4\mu\text{seg}$ .



**Cálculo de erro** calcula a diferença de amplitude entre as entradas representadas nas extremidades do bloco e a entrada central. Isto permite conhecer se existe erro de sincronismo, e automaticamente corrigi-lo.

- Se a entrada central (com um *delay*) for a entrada com maior amplitude em módulo no instante de amostragem, então não existe erro;
- Se a entrada que passa pelos dois *delays*, ou seja a mais atrasada, for a entrada com maior amplitude no instante de amostragem, então é necessário atrasar o instante de amostragem. Assim, o erro calculado é positivo para que o próximo instante de amostragem seja maior ( $T + \Delta t$ ), em que  $\Delta t$  é o passo constante a dar para tentar acertar a sincronização;
- Se a entrada que vem diretamente da saída do integrador, ou seja a mais adiantada, for a entrada com maior amplitude, então é necessário adiantar o instante de amostragem. Assim, o erro calculado é negativo para o que o próximo instante de amostragem seja menor ( $T - \Delta t$ ).

**Detetor** aproveita as amostras retiradas do instante de amostragem da saída do integrador com um *delay*, isto é, a saída que o bloco de cálculo de erro considera como a mais síncrona. O Detetor também contém o bloco que efetua a descodificação, convertendo o sinal de NRZI para símbolos de informação.

**Delay** atrasa o sinal à sua entrada. O valor deste atraso é determinado com o recurso a testes, de forma a sincronizar de forma eficaz e rápida<sup>4</sup> o sistema realimentado.

Estamos assim novamente perante um sistema que também é realimentado, prevenindo a ocorrência de desestabilização do recuperador de relógio.

Também por ser um sistema realimentado, necessita de um tempo de estabilização inicial por forma a capturar o relógio a fixar no ritmo de transmissão dos símbolos. No entanto, este facto não é preocupante para o projeto em questão, pois como visto no capítulo anterior, a norma obriga a um preâmbulo inicial propositado para este tipo de aplicações.

Ainda com o objetivo de não prejudicar os circuitos recuperadores de relógio, a norma EN 12795 (que regula a camada MAC), não permite a ocorrência de mais do que 5 símbolos 1's consecutivos, pois como a codificação de 1's em NRZI é a não transição de fase, um acumular de 1's seguidos superior ao referido podia provocar uma desregulação na recuperação do relógio.

---

<sup>4</sup>a referência a “rápido” deve-se à comparação do tempo de estabilização do sistema com a duração máxima prevista para o efeito, ou seja, a duração do preâmbulo determinado pela norma EN 12253.

### 3.5 Esquema da Simulação

De forma a conseguir testar e ilustrar o processo de receção de sinais em comunicações DSRC, as próximas secções irão desencadear a explicação e os resultados da simulação usada onde estão incluídos os vários métodos de desmodulação de sinal BPSK. Assim, com base nas desvantagens e vantagens de cada método, será possível escolher qual o melhor método a usar no recetor na fase de implementação.

Na Figura 3.10 é ilustrado o diagrama de blocos que serve de base para a simulação que será apresentada.

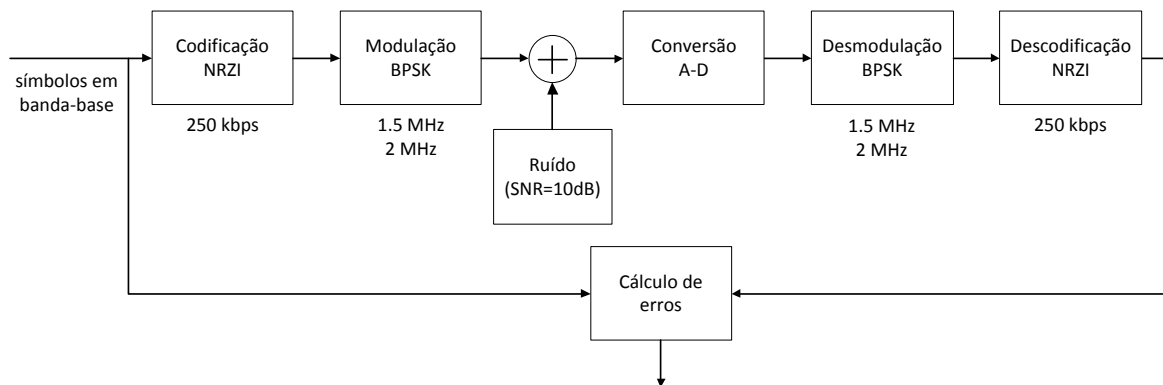


Figura 3.10: Diagrama de blocos da simulação usada.

Com base no diagrama, a simulação é efetuada da seguinte forma:

- Primeiro, os blocos necessários para a preparação do sistema a simular:

É gerado um sinal com codificação NRZI em banda-base e com modulação BPSK a 1.5 MHz ou 2 MHz (blocos Codificação NRZI e Modulação BPSK);

De seguida, é adicionado ruído ao sinal gerado, de forma a que a relação entre o sinal e o ruído (SNR) seja de 10 dB (bloco Ruído);

- Blocos que simulam o recetor:

O sinal passa por um andar que visa simular a conversão analógico-digital (bloco Conversão A-D);

No processo de receção BPSK são basicamente tratados os dois métodos apresentados anteriormente (bloco Desmodulação BPSK);

Finalmente, é simulado o processo de recuperação de relógio de transmissão, de forma a amostrar e descodificar o sinal em banda-base (bloco Descodificação NRZI).

### 3.6 Simulação de codificação NRZI e Modulação BPSK

Como apresentado na secção 2.1, os parâmetros a ter em conta para o processo de transmissão, são o tipo de codificação, tipo de modulação, taxa de transmissão, frequência de sub-portadora e especificações do preâmbulo.

No seguimento da simulação, foi considerado o uso da sub-portadora de 2 MHz na modulação BPSK. No entanto, considera-se que para a opção da sub-portadora de 1.5 MHz, o processo é análogo, como é o caso da projeção dos filtros a usar.

A taxa de transmissão em banda-base é 250 kbps e o tempo de preâmbulo para transmissão de sub-portadora pura é de  $32 \mu s$  (pela norma é permitido de 32 a  $36 \mu s$ ). Também é gerada a parte de preâmbulo a seguir a esta, especificada como a transmissão de oito 0's, ou seja, oito transições de fase espaçadas de  $4 \mu s$ .

A frequência de amostragem usada é de 16 MHz, escolhida para ser o melhor partido entre os filtros não necessitarem de ordens altas e para prevenir a ocorrência do efeito de *aliasing*.

A Figura 3.11 mostra o preâmbulo criado no MATLAB com as especificações referidas anteriormente. É possível visualizar parte dos  $32 \mu s$  de transmissão de sub-portadora pura e parte dos oito 0's transmitidos.

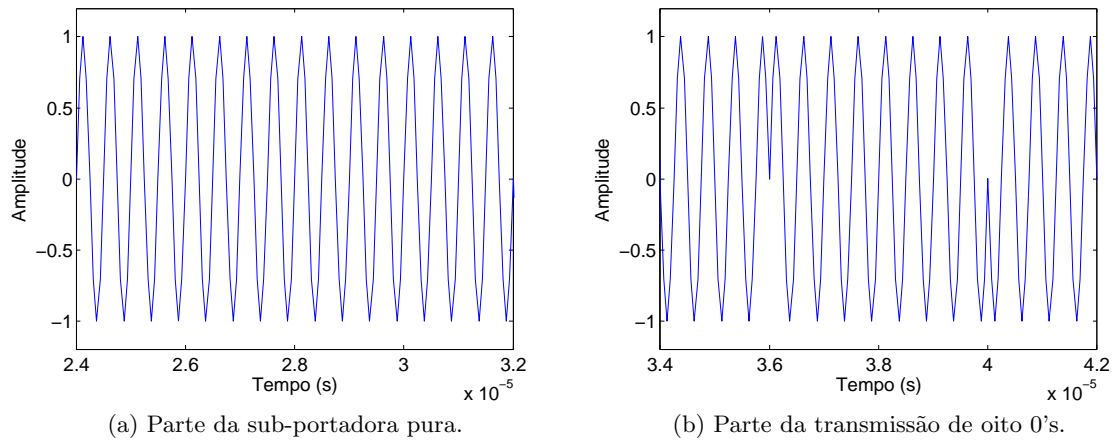


Figura 3.11: Preâmbulo modulado a 2 MHz.

São depois gerados *bits* de forma pseudo-aleatória para representar a trama de informação. Nesta simulação foram gerados 1000 *bits* e o espectro de frequências resultante, está ilustrado na Figura 3.12.

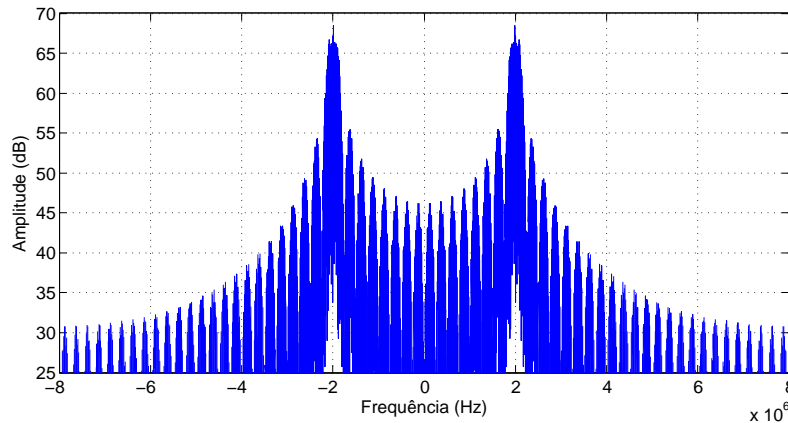


Figura 3.12: Espectro de frequência do sinal transmitido a 2 MHz.

Com maior ampliação da Figura 3.12 é possível confirmar que a largura de banda do lóbulo principal do sinal transmitido é de  $2R = 2 \cdot 250 \text{ kbps} = 500 \text{ KHz}$ . Esta largura de banda é importante para a implementação dos filtros nas próximas fases da simulação.

### 3.7 Introdução de ruído

Para a introdução de ruído no sinal, foi usado um bloco da ferramenta Simulink do MATLAB designado “AWGN Channel”. Este bloco adiciona ruído Gaussiano branco ao sinal de entrada. Na sua utilização é possível usar vários modos para especificar o ruído a adicionar ao sinal. Um deles, o que foi usado, é através da especificação da relação SNR. O bloco fica ainda a necessitar de mais um campo, que é a potência do sinal de entrada, calculado da seguinte forma:

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2 \quad (3.33)$$

O SNR calculado internamente no bloco, de forma a colocar no sinal de saída o SNR de 10 dB pretendido, relaciona as potências do sinal e do ruído da seguinte forma:

$$\text{SNR}_{dB} = 10 \log \frac{P_{\text{sinal}}}{P_{\text{ruído}}} = P_{\text{sinal},dB} - P_{\text{ruído},dB} \quad (3.34)$$

De notar que esta relação SNR entre o sinal e ruído é calculada para toda a largura do sinal (dada pela frequência de amostragem utilizada), e não apenas na banda de interesse. Numa maior aproximação à realidade e ao especificado na norma reguladora, a relação deverá ser calculada tendo em conta as potências do ruído e do sinal dentro da banda de interesse. No capítulo dos resultados será feito dessa forma. A comparação do espectro de frequências entre os sinais à entrada e à saída do bloco “AWGN Channel” é representada na Figura 3.13.

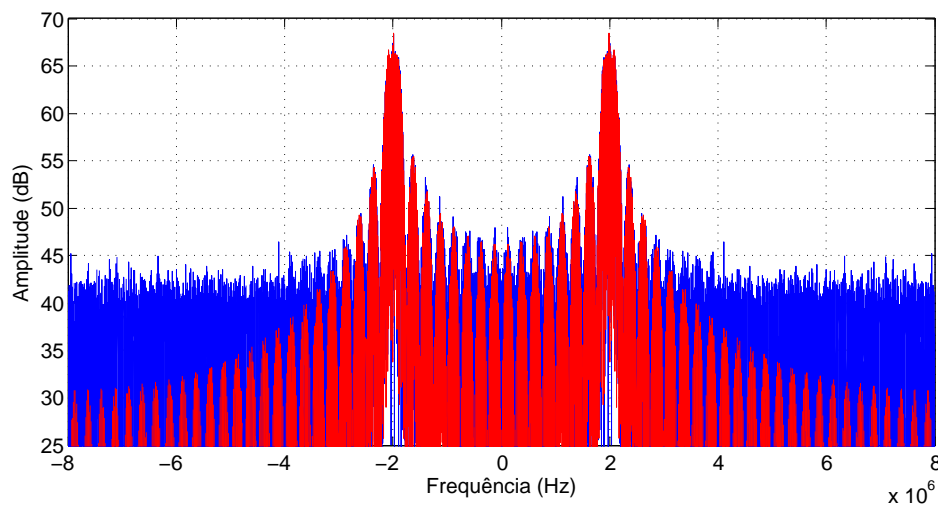


Figura 3.13: Espectros de frequências dos sinais com e sem ruído. A vermelho, o sinal sem ruído. A azul, o sinal com ruído.

### 3.8 Andar de conversão analógico-digital

Este passo visa a simulação do andar a ser construído no processo de implementação, que terá um filtro passa-banda analógico geral (deixa passar as bandas de interesse tanto para quando o sinal é transmitido a 1.5 MHz ou a 2 MHz) e um conversor analógico-digital (ADC).

Para que o filtro deixe passar a largura de banda de 500 KHz, tanto com sub-portadora de 1.5 MHz como com sub-portadora de 2 MHz terá que ter uma largura iniciada em 1.25 MHz até aos 2.25 MHz. A representação da resposta em frequência do filtro implementado no MATLAB é visível na Figura 3.14. Este filtro tem o corte a 3 dB às frequências de 1.23 MHz e de 2.28 MHz.

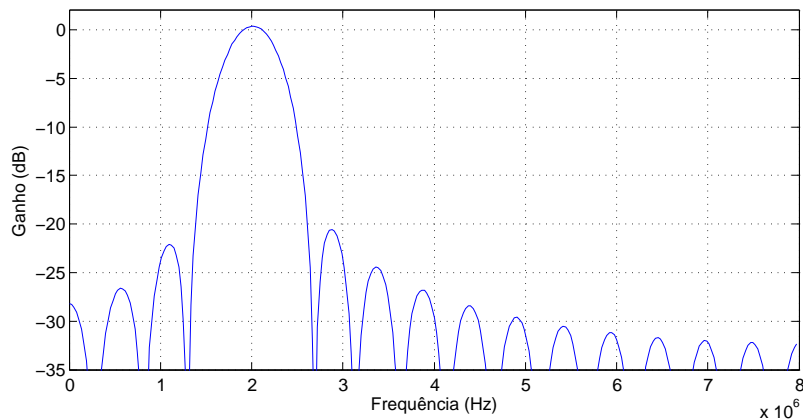


Figura 3.14: Resposta em frequência do filtro passa-banda do andar de conversão.

Para a simulação da conversão analógica para digital, é usado uma ADC de 10 *bits*, à semelhança da ADC a usar na implementação. O espectro de frequências após o filtro passa-banda e após a conversão analógica para digital é visível na Figura 3.15.

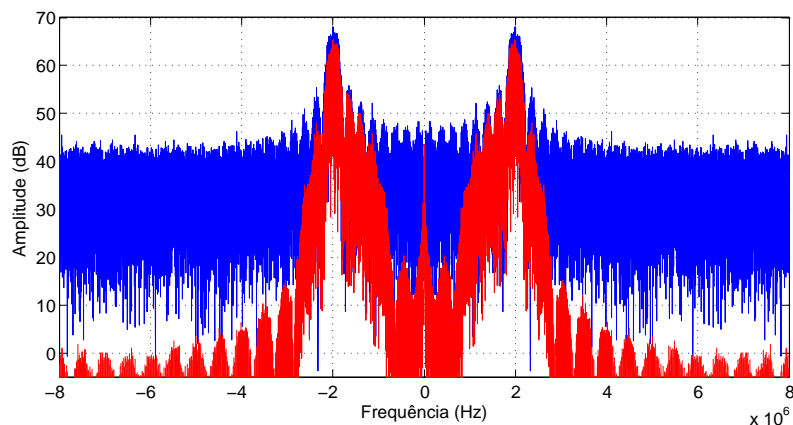


Figura 3.15: Espectro de frequências após o filtro passa-banda geral e ADC. A vermelho, o espectro depois do filtro e ADC. A azul, o espectro do sinal inicial com ruído.

Prevendo-se que a amplitude do sinal pode ultrapassar a gama de leitura da ADC, saturando-a, o espectro da Figura 3.16 ilustra o espectro de frequências para esse caso.

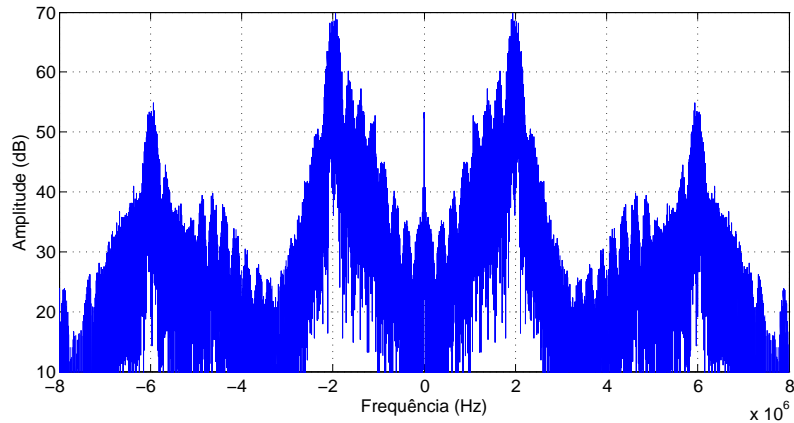


Figura 3.16: Espectro de frequências após o filtro passa-banda geral e ADC com saturação na ADC.

Como a onda é saturada à entrada, por ultrapassar a gama da ADC, o sinal à saída tem aproximadamente a forma de uma onda quadrada. Assim, o espectro é agora composto por outros picos de frequências (na Figura 3.16 em -6 MHz e 6 MHz). No entanto, estes picos são atenuados à entrada do desmodulador de BPSK com um novo filtro passa-banda.

### 3.9 Simulação desmodulador BPSK

Para esta etapa da simulação, a desmodulação BPSK, temos essencialmente três blocos: um filtro passa-banda, um recuperador de portadora e o desmodulador em si. O filtro passa-banda é agora mais estreito, pois é dependente da frequência da sub-portadora usada.

#### 3.9.1 Filtro passa-banda

Como nesta simulação estamos a usar uma sub-portadora de 2 MHz, o filtro representado na Figura 3.17 está desenhado para essa especificação. Este filtro tem o corte a 3 dB às frequências de 1.7 MHz e de 2.3 MHz aproximadamente.

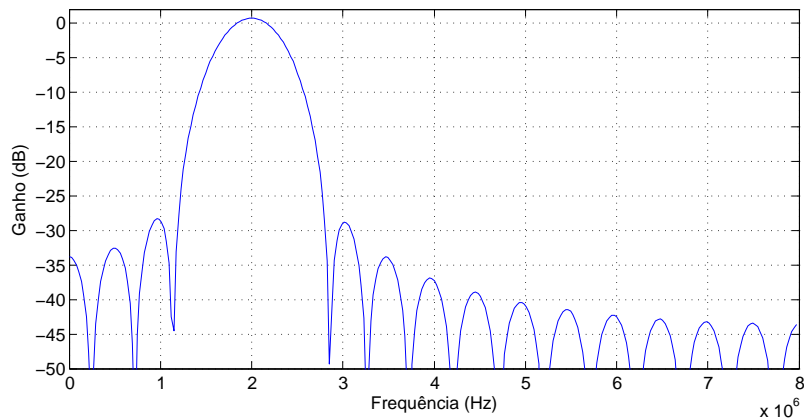


Figura 3.17: Resposta em frequência do filtro passa-banda do desmodulador BPSK.

### 3.9.2 Simulação da extração da portadora pelo quadrado do sinal recebido

O diagrama de blocos da Figura 3.18 representa as funções usadas na simulação deste método.

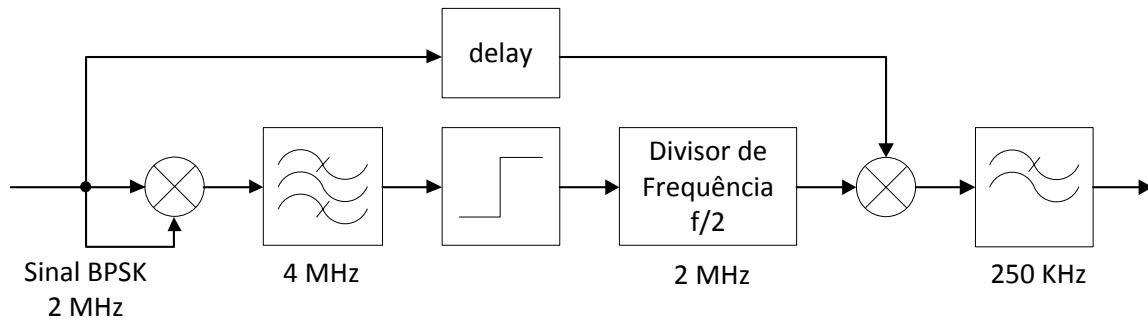


Figura 3.18: Diagrama de blocos do método de extração de portadora pelo quadrado do sinal recebido.

#### 3.9.2.1 Filtro passa-banda a 4 MHz

Este filtro passa-banda, logo após a multiplicação – quadrado do sinal BPSK recebido, tem o objetivo de extrair a sub-portadora 4 MHz. Como visto na sub-seção 3.3.1, com esta multiplicação o sinal modulante em banda-base é eliminado, ficando apenas a sub-portadora ao dobro da frequência.

A Figura 3.19 mostra a resposta em frequência do filtro projetado no MATLAB.

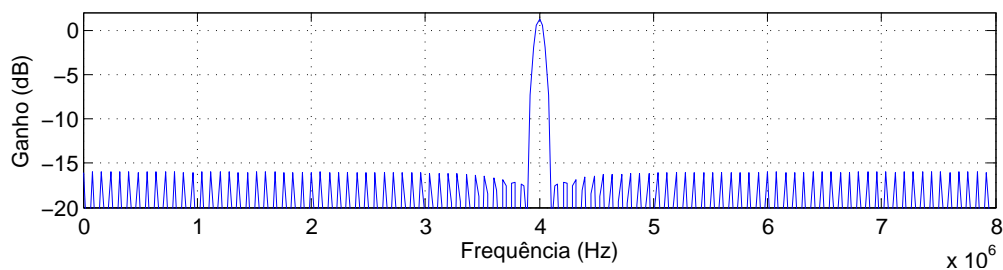


Figura 3.19: Resposta em frequência do filtro passa-banda de 4 MHz.

Este filtro com número de ordem de cerca de 200 (corte de 3 dB a 3.94 MHz e 4.06 MHz) é um dos principais entraves para implementar este método de desmodulação. Por ser um filtro passa-banda com largura de banda de passagem bastante estreita, para fazer passar a sub-portadora o mais pura possível, a sua implementação exige que tenha um elevado número de coeficientes.

Uma das formas de tentar minimizar a ordem do filtro, seria diminuir a frequência de amostragem, decimando o sinal nesta etapa. À frequência de trabalho de 4 MHz, o máximo que eventualmente se conseguiria diminuir era para 12 MHz, para evitar efeito de *aliasing* ( $12 \text{ MHz} / 2 = 6 \text{ MHz}$  próximo de 4 MHz). Mas, mesmo a esta frequência de amostragem de 12 MHz, com uma largura de banda do filtro de cerca de 200 KHz em comparação com  $f_a/2 = 6 \text{ MHz}$ , a relação continua a ser elevada e o filtro ainda necessitaria de um elevado número de ordem.

Outro facto é que este número de ordem do filtro (superior a 100) ultrapassa em larga escala o número de multiplicadores dedicados na FPGA, lembrando ainda que parte destes vão também ser necessários para outros filtros que constituem a restante parte do recetor.

### 3.9.2.2 Saturar o sinal para criar uma onda quadrada

O bloco que se segue no diagrama da Figura 3.18 é um bloco que tem como finalidade tornar o sinal numa onda quadrada. Este passo intermédio entre o filtro passa-banda e o divisor de frequência tem o objetivo de simplificar o sinal para que seja possível depois dividi-lo em frequência.

Na Figura 3.20 é ilustrado o espectro do sinal a 4 MHz em forma de onda quadrada.

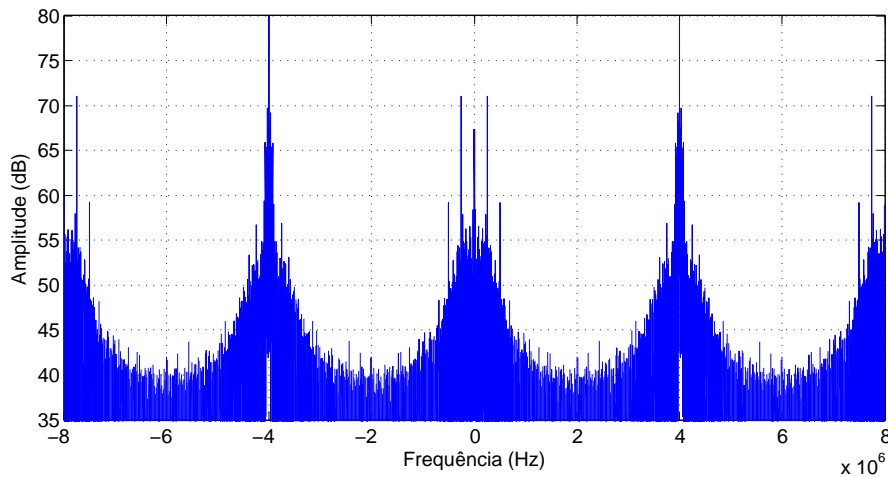


Figura 3.20: Espectro de frequências da subportadora a 4 MHz em forma de onda quadrada.

### 3.9.2.3 Divisor de frequência por 2

Tal como o nome deste bloco indica, a sua finalidade é dividir a frequência do sinal para metade. Isto é possível de efetuar de forma simples, pois o sinal de entrada está na forma de uma onda quadrada.

Para fazer a divisão, basta considerar que o sinal de saída transitará de fase (de -1 para 1, ou de 1 para -1, conforme o nível em que o sinal se encontrar no instante de mudança) sempre que no sinal de entrada tiver sido percorrido um período do sinal. Isto é, sempre que no sinal de entrada ocorrer um *rising edge* (ou de forma análoga sempre que houver um *falling edge*, qualquer das formas pode ser considerada).

De forma a exemplificar este processo, na Figura 3.21 é possível visualizar a comparação entre o sinal de entrada e o sinal de saída deste bloco. Note-se o facto de que na simulação foi considerado o *falling edge* para a transição do nível do sinal de saída.



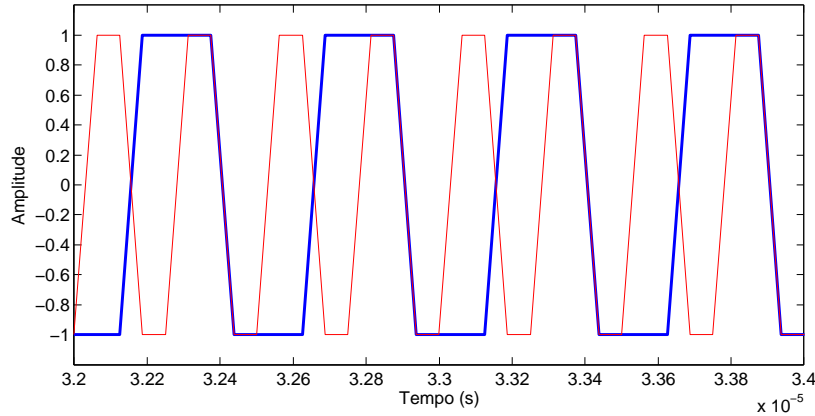


Figura 3.21: Exemplo de intervalo de tempo das sub-portadoras. A vermelho, sub-portadora a 4 MHz. A azul, sub-portadora depois de dividida, a 2 MHz.

Depois deste bloco, o espectro de frequências resultante é o representado na Figura 3.22.

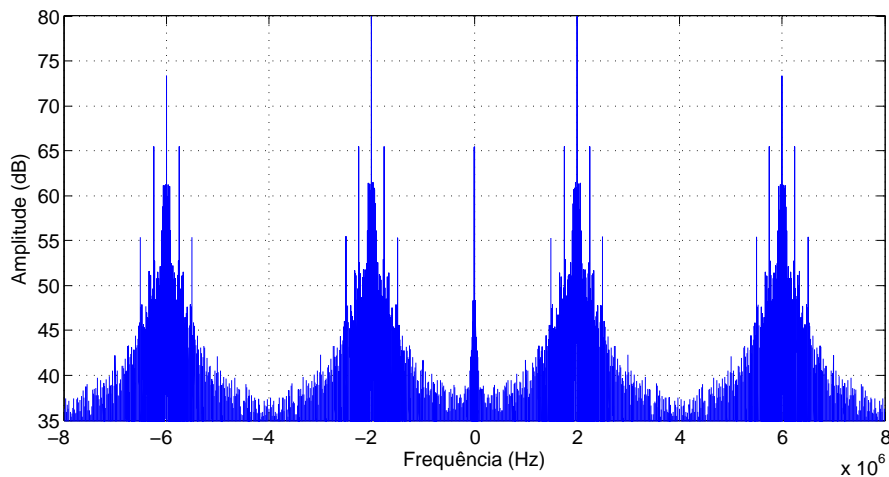


Figura 3.22: Imagem do MATLAB do espectro de frequências da subportadora a 2 MHz em forma de onda quadrada.

#### 3.9.2.4 Bloco de atraso “delay”

A necessidade deste bloco é outra desvantagem face ao método *Costas loop* que será simulado de seguida. Por haverem dois ramos em paralelo que precisam de estar sincronizados no momento da multiplicação entre eles, é necessário criar um bloco de atraso para compensar o atraso provocado no processo da extração da portadora.

O atraso resultante do ramo de extração de portadora é provocado essencialmente pelo filtro passa-banda. O número de amostras de atraso é igual a  $N+1$  (em que  $N$  é o número de ordem do filtro).

Por ser possível prever o número de amostras de atraso, a sincronização entre os sinais é efetuada atrasando o ramo que contém o sinal a desmodular.

### 3.9.2.5 Filtro passa-baixo para banda-base

Depois de multiplicar a portadora pelo sinal BPSK, o sinal modulante fica desmodulado em banda-base. No entanto, depois da multiplicação é criada uma componente ao dobro da frequência de modulação (4 MHz) que precisa de ser filtrada por um filtro passa-baixo.

O sinal depois de passar pelo filtro passa-baixo fica assim efetivamente desmodulado. Na Figura 3.23 é ilustrado o preâmbulo desmodulado.

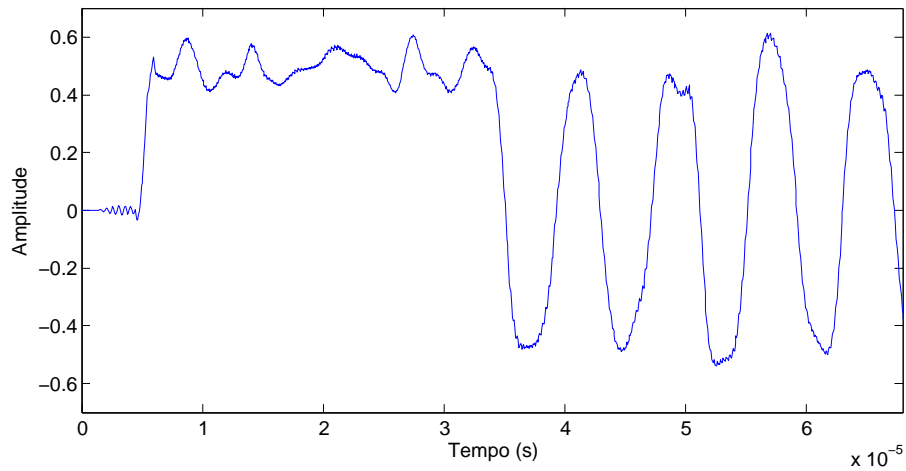


Figura 3.23: Preâmbulo desmodulado, em banda-base.

### 3.9.3 Simulação *Costas loop*

O diagrama de blocos da Figura 3.24 representa as funções usadas na simulação deste método. Esta figura é semelhante à Figura 3.6, repetida aqui por conveniência.

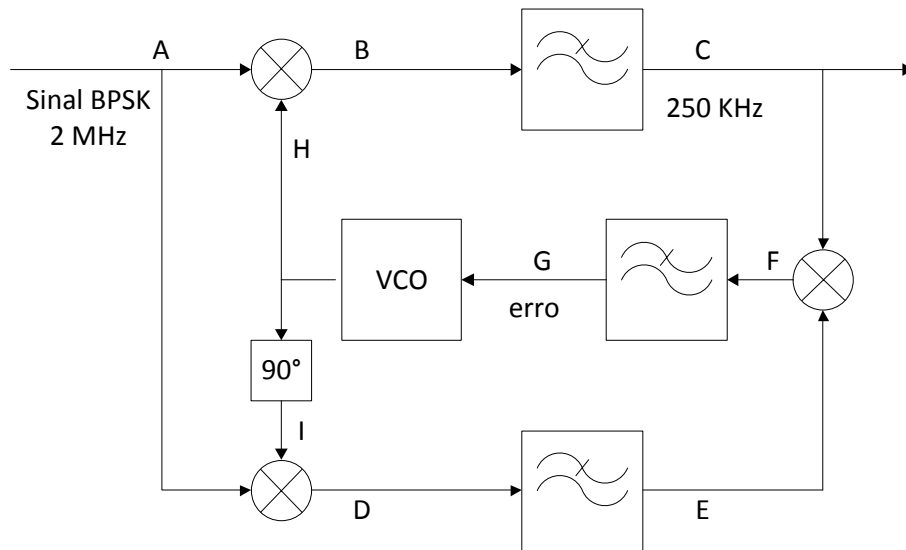


Figura 3.24: Diagrama de blocos do método *Costas loop*.

### 3.9.3.1 Filtros passa-baixo (C e E)

Considerando que a sub-portadora em H está em fase com o sinal BPSK em A, na saída do multiplicador em B é produzido o sinal desmodulado, mas com uma componente inserida a 4 MHz (ao dobro da frequência das entradas no multiplicador). Esta componente é filtrada pelo filtro passa-baixo, que produzirá o sinal desmodulado efetivo em C. O filtro passa-baixo no ramo em quadratura é projetado de forma análoga.

Este filtro não necessita de ser muito preciso nem estreito, pois tem como único objetivo filtrar essa componente espectral ao dobro da frequência do sinal modulado. Assim, é um filtro com ordem relativamente baixa, quando comparamos com a ordem de filtros usados no método anterior. A resposta em frequência do filtro implementado no MATLAB para a simulação é representado na Figura 3.25. O filtro tem ordem de 5 e corte 3 dB a 1.5 MHz aproximadamente.

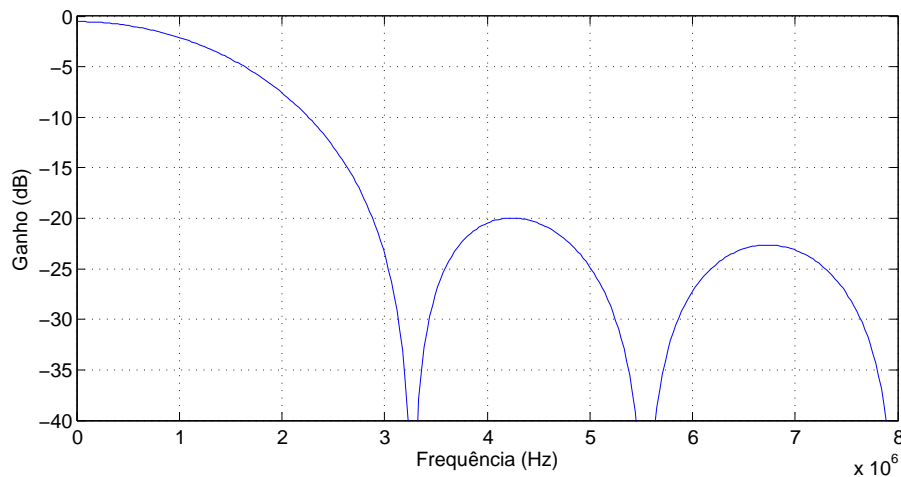


Figura 3.25: Resposta em frequência do filtro passa-baixo em C e E.

Pela Figura 3.25 verifica-se que o corte tanto a 4 MHz como a 3 MHz (para o caso em que a sub-portadora é de 1.5 MHz) é de aproximadamente -20 dB. Assim, este filtro está preparado para ambas as opções de sub-portadora. A uso destes simples filtros contrasta com o método analisado anteriormente em que eram necessários filtros passa-banda de elevada ordem e projetados de forma separada para cada uma das frequências da sub-portadora.

### 3.9.3.2 Correção do VCO através da realimentação

Como referido na sub-seção 3.3.2, o filtro passa-baixo inserido dentro da malha serve essencialmente para filtrar altas frequências que inevitavelmente se propagam à saída dos filtros anteriores, por estes não serem completamente ideais. Este filtro torna assim a amplitude do erro a entrar no VCO mais estável, condicionando no entanto a rapidez do *loop*.

Um exemplo da amplitude de erro à entrada do VCO é ilustrada na Figura 3.26. Trata-se do período de tempo correspondente ao preâmbulo inicial da trama, sendo assim possível de analisar a correção inicial que o VCO tem de fazer para sincronizar a sub-portadora com o sinal BPSK recebido. A correção inicial está dentro do período de sub-portadora pura do preâmbulo inicial, 32  $\mu s$  (Figura 3.27).

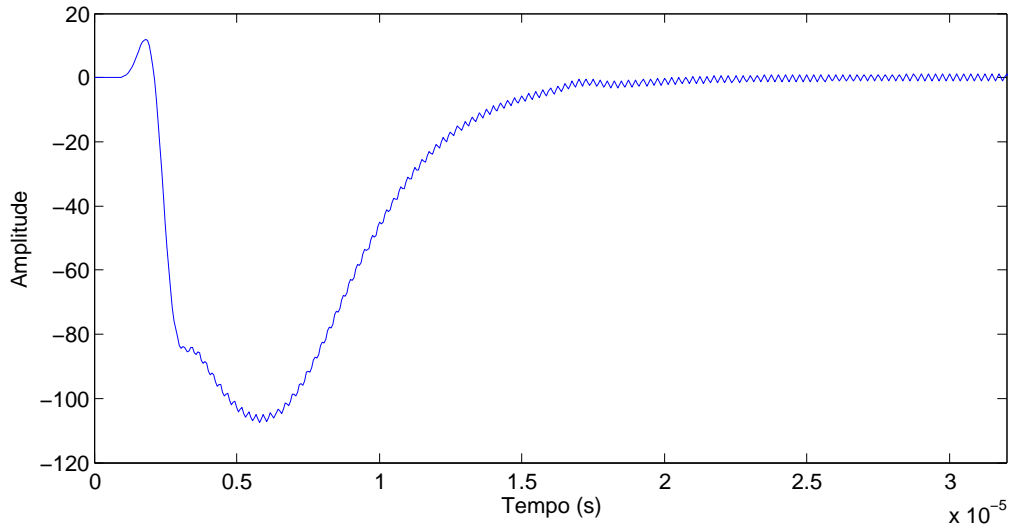


Figura 3.26: Erro à entrada do VCO durante o período de preâmbulo.

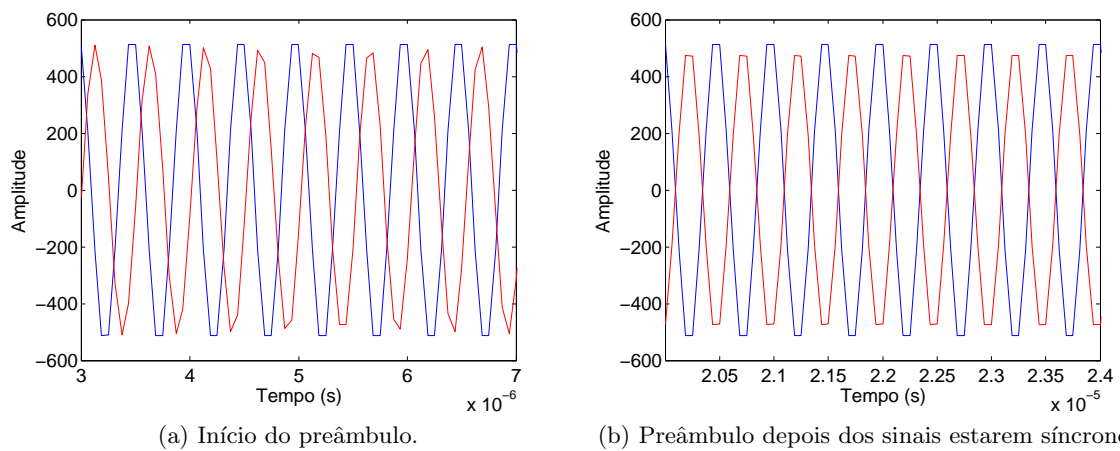


Figura 3.27: Sincronização entre o sinal recebido e a sub-portadora à saída do VCO. A vermelho, sub-portadora gerada pelo VCO. A azul, sinal BPSK recebido.

### 3.9.3.3 Saída do *Costas loop* – sinal desmodulado

À saída do filtro passa-baixo em C (Figura 3.24) temos o sinal em banda-base efetivamente desmodulado. Na Figura 3.28 é ilustrado o preâmbulo desmodulado. Nesta figura, é possível verificar, através da ascensão da amplitude em módulo, o tempo de estabilização do sistema durante o período do preâmbulo.

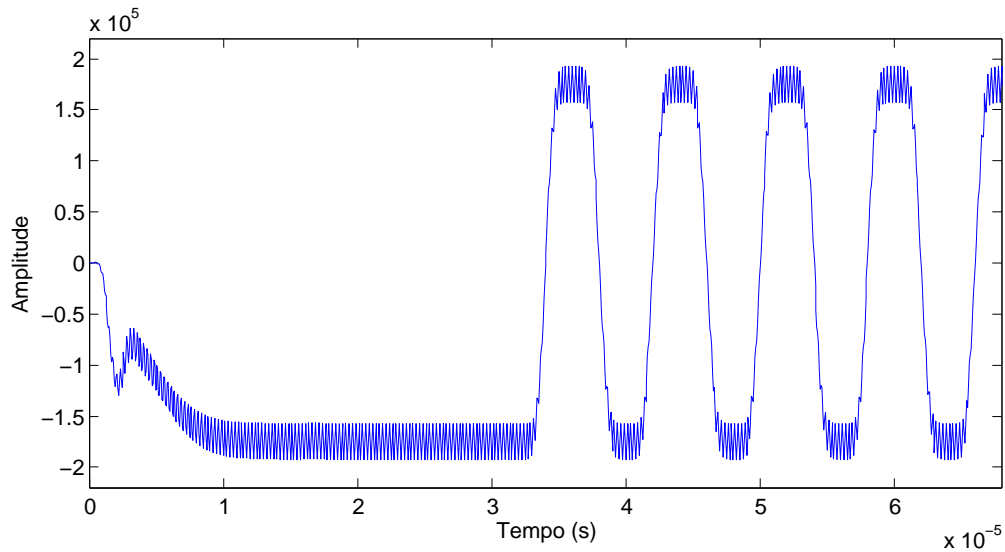


Figura 3.28: Preâmbulo desmodulado, em banda-base, com referência à estabilização inicial do sistema.

### 3.9.4 Especificação do método de desmodulação a implementar

Com base nas simulações anteriores dos dois métodos (extração de portadora pelo quadrado do sinal recebido e *Costas loop*), pode-se resumir alguns pontos de comparação:

- O primeiro método necessita de filtros passa-banda de elevada ordem, e de um para cada opção de sub-portadora (1.5 MHz e 2 MHz). O método *Costas loop* não precisa destes filtros. Mesmo com o *Costas loop* a necessitar de mais filtros passa-baixo, estes são na generalidade de baixa ordem e projetados para permitir a receção de sinais de qualquer opção de sub-portadora.
- O método *Costas loop* necessita de um tempo de estabilização do sistema superior ao primeiro método (comparando a Figura 3.28 com a Figura 3.23). No entanto, desde que seja garantido que este tempo de estabilização não ultrapasse o tempo de preâmbulo, este ponto não é considerado uma desvantagem.

Para comprovar que o tempo de estabilização se mantém dentro do período de tempo de sub-portadora pura, a Figura 3.29 ilustra o gráfico de vários resultados sobrepostos de desmodulação de sinais de entrada com erros de fases diferentes à entrada do desmodulador.

- O primeiro método obriga a calibração no que cabe à sincronização entre a sub-portadora extraída e o sinal a desmodular. Esta sincronização é conseguida, criando um bloco de atraso “artificial”, mas que pode necessitar de ser diferente para cada opção de frequência da sub-portadora. Pode-se considerar uma vantagem o facto do método *Costas loop* não necessitar deste tipo de calibração, pois a sincronização é efetuada automaticamente pela realimentação.

Perante os pontos referidos anteriormente, o método de desmodulação que vai ser utilizado na fase de implementação deste trabalho será o *Costas loop*.

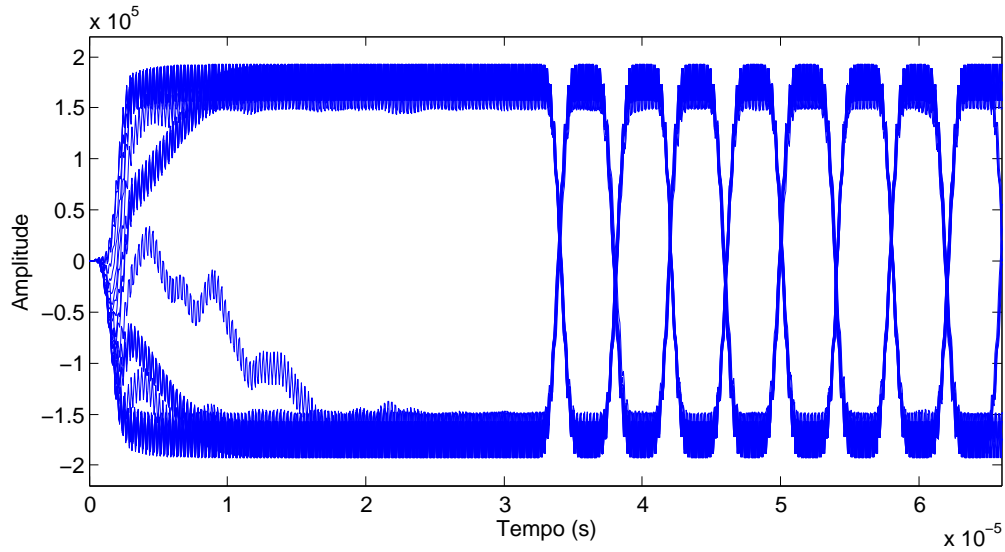


Figura 3.29: Sobreposição de preâmbulos desmodulados com erros de fases diferentes na entrada do desmodulador. Verifica-se que o sinal fica sempre estável antes de antigir o final do período de sub-portadora pura (aos  $32\mu\text{s}$ ).

### 3.10 Simulação do recuperador do relógio de transmissão em banda-base

O diagrama de blocos da Figura 3.30 representa os blocos usados na simulação do recuperador do relógio de transmissão em banda-base.

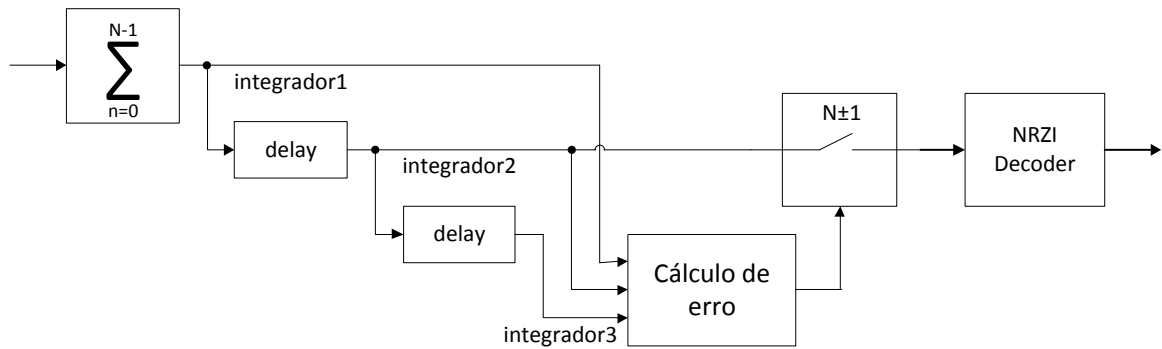


Figura 3.30: Diagrama de blocos do recuperador do relógio de transmissão em banda-base.

O integrador finito representado na Figura 3.30 é um somador das últimas  $N$  amostras do sinal em cada instante.  $N$  é o número de amostras por símbolo, obtido através de  $N = fa/R$ , em que  $fa$  é a frequência de amostragem e  $R$  é o ritmo de transmissão dos símbolos.

Deste integrador, *integrador1*, são gerados mais dois sinais. Nesta simulação *integrador2* é atrasado  $N/4$  amostras e *integrador3* é atrasado  $N/2$  amostras. O sinal a amostrar é o *integrador2*, pois é este que o bloco “Cálculo de erro” considera que está em fase com o ritmo de transmissão a recuperar.

De modo a corrigir a amostragem, o bloco “Cálculo de erro” dá indicação de adiantar ou atrasar o relógio recuperado em passos de  $N/16$  amostras, ou em caso de não existir erro, deixa o período de amostragem intacto:

- Se *integrador1*, o integrador mais adiantado, for a entrada com maior amplitude no instante de amostragem, então são adiantadas  $N/16$  amostras;
- Se *integrador2* for a entrada com maior amplitude em módulo no instante de amostragem, então não existe erro;
- Se *integrador3*, o integrador mais atrasado, for a entrada com maior amplitude no instante de amostragem, então são atrasadas  $N/16$  amostras.

Depois, o bloco “NRZI Decoder” retira a codificação de NRZI do sinal amostrado.

De forma a exemplificar este processo de recuperação do relógio de transmissão, a Figura 3.31 mostra os três integradores e o instante em que o sinal foi amostrado, com o objetivo de exemplificar a correção automática de amostragem. O período de tempo visível é, mais uma vez, o tempo de preâmbulo no início da trama.

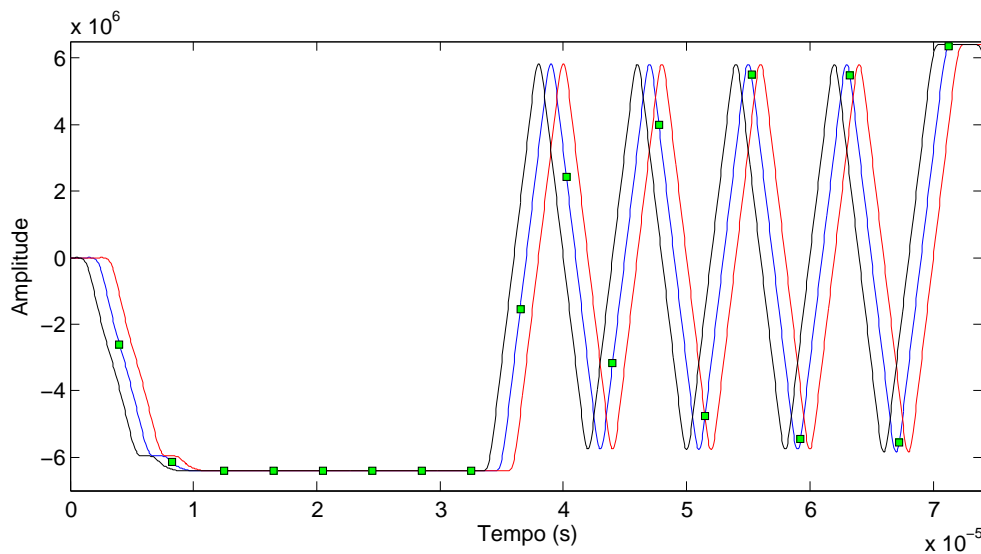


Figura 3.31: Preâmbulo desmodulado e integrado. São visíveis os três integradores em que o integrador central é o relativo à amostragem. Os pontos correspondem aos instantes em que ocorre a amostragem.

Pela Figura 3.31 é possível perceber o porquê da existência dos símbolos 0's de preâmbulo. Estes 0's têm o objetivo de dar ao recetor tempo de conseguir recuperar o relógio de transmissão.

Na primeira mudança de fase aos  $32 \mu s$ , o instante de amostragem tem erro, pois está sensivelmente próximo da origem. Como o *integrador1* (o integrador adiantado) tem maior amplitude absoluta que o *integrador2*, o próximo instante de amostragem vai acontecer  $N/16$  amostras adiantadas. A correção acontecerá constantemente até o instante de amostragem conseguir “encontrar” o pico de amplitude.

É visível que no fim dos oito 0's de preâmbulo, o relógio já está praticamente síncrono, e o recetor está em condições de conseguir amostrar os símbolos com eficácia.

### 3.11 Resultados quantitativos da simulação

Através do bloco de cálculo de erros representado na Figura 3.10, foi possível quantificar a existência ou não de erros entre a sequência enviada e a sequência recebida.

Embora o MATLAB não seja a ferramenta ideal para fazer testes exaustivos com um número elevado de *bits* a enviar, devido às limitações de processamento em matrizes de tamanho bastante elevado, foi possível realizar testes na ordem de  $10^5$  *bits* enviados. Não foram detetados quaisquer erros. Recorde-se que o SNR era de 10 dB.

Assim, foi considerado que haviam condições para passar à etapa de implementação, aplicando os métodos até aqui estudados e simulados.



## Capítulo 4

# Implementação do Recetor

### 4.1 Esquema do Recetor

Como já referido anteriormente, para a implementação do presente recetor do *Road Side Unit (RSU)* para comunicações DSRC vão ser usadas técnicas de *Software Defined Radio (SDR)*. Das várias arquiteturas de digitalização apresentadas na secção 2.2 será utilizada a digitalização em IF (recorde-se novamente através da Figura 4.1). Esta arquitetura reúne algumas das vantagens do SDR onde estas podem ser aplicadas no sistema através de técnicas digitais em banda-base e IF. No entanto, continua a ser necessário *hardware* analógico nas frequências de RF, onde será utilizado um *front-end* já existente.

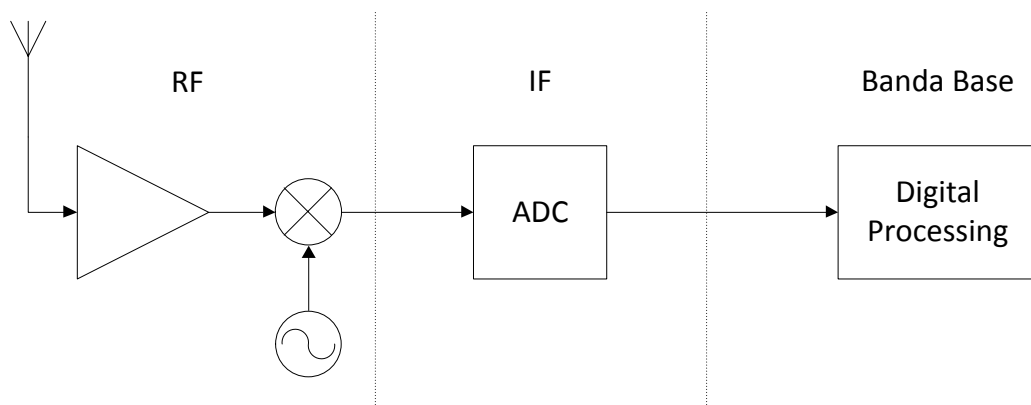


Figura 4.1: Arquitetura de digitalização em IF.

Com base nesta arquitetura, a Figura 4.2 apresenta o esquema geral do recetor implementado. A implementação será dividida em três secções – estudo das características dos sinais obtidos do *front-end* existente de 5.8 GHz, construção de um andar de conversão analógico-digital dos sinais a 1.5 MHz ou 2 MHz, e por fim, implementação do desmodulador na FPGA.

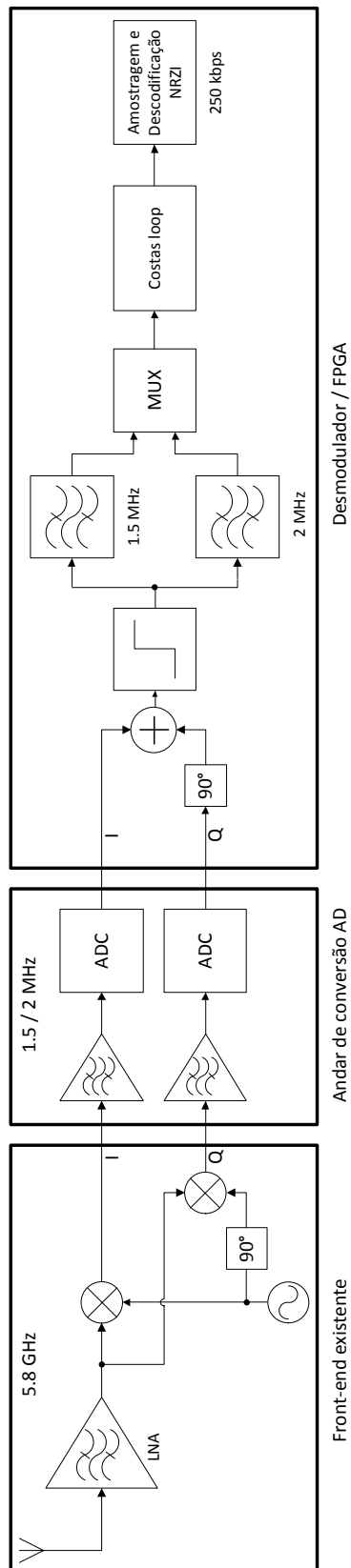


Figura 4.2: Esquema do recetor implementado.

## 4.2 Front-end existente de 5.8 GHz

Este *front-end* para comunicações DSRC é parte constituinte de um RSU existente no Instituto de Telecomunicações. Os valores apresentados de seguida foram dados a conhecer a este trabalho como enunciado do problema a resolver.

A Tabela 4.1 apresenta dados das potências referentes a vários pontos na transmissão DSRC – do sinal enviado pelo RSU e de seguida retornado pelo OBU. O balanço termina com o cálculo da potência à entrada do *front-end*, isto é, depois da antena do recetor.

Tabela 4.1: Ganhos/perdas e balanço de potências em vários pontos da transmissão DSRC

	Ganho (dB)	Balanço (dBm)
PIRE (transmissão) do RSU		+33
Perdas em espaço livre (10 metros)	-68	-35
Atenuação para-brisas + condições meteorológicas	-3	-38
Ganho Antena OBU	+7	-31
Offset antena OBU	-2	-33
Perdas no modulador do OBU	-10	-43
Ganho Antena OBU	+7	-36
Offset antena OBU	-2	-38
Atenuação para-brisas + condições meteorológicas	-3	-41
Perdas em espaço livre (10 metros)	-68	-109
Ganho da antena do RSU	+14	<b>-95</b>

Com uma potência de sinal de 95 dBm (0.316pW) e admitindo a antena do recetor adaptada a 50  $\Omega$ , o equivalente de Thèvenin tem um gerador cuja amplitude de pico é:

$$P = \frac{1}{8} \frac{V^2}{R} \Leftrightarrow V = \sqrt{8(P \cdot R)} \approx 11\mu V \quad (4.1)$$

A Figura 4.3 ilustra o diagrama do *front-end* existente. Este é composto por vários andares de ganho e de filtragem, com três LNAs, dois filtros passa-banda, um divisor de potência e dois misturadores (um por cada ramo – em fase e em quadratura).

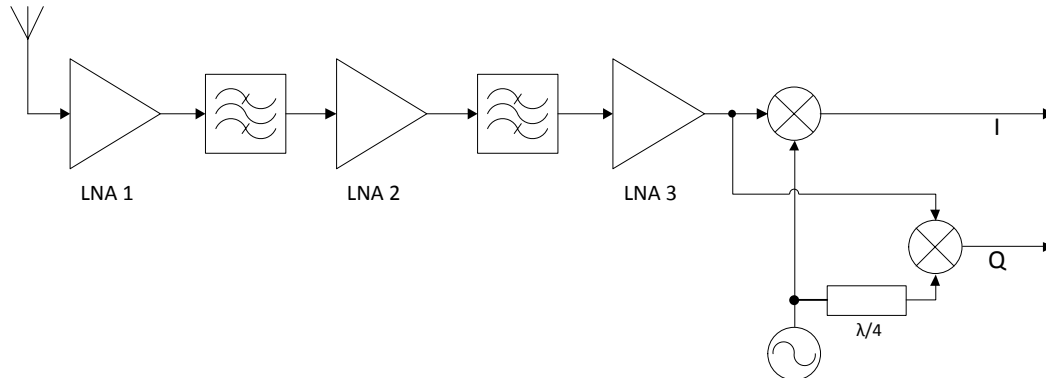


Figura 4.3: Diagrama do front-end existente.

Através da Tabela 4.2, são possíveis de analisar os ganhos e interferências de ruído, *Noise Figure (NF)*, no decorrer dos vários andares do *front-end*. Isto leva ao resultado final de qual o ganho acumulado ao longo da cadeia.

Tabela 4.2: Ganhos/perdas e *Noise Figure* dos vários componentes do *front-end*

Componente	NF componente (dB/linear)	Ganho componente (dB/linear)	NF acumulado (dB/linear)	Ganho acumulado (dB/linear)
LNA HMC320	2.6/1.82	13/20	2.6/1.82	13/20
LNA HMC320	2.6/1.82	13/20	2.7/1.86	26/400
LNA HMC320	2.6/1.82	13/20	2.7/1.86	39/8000
Divisor de potência	3/2	-3/0.5	2.7/1.86	36/4000
Misturador HMC218	8/6.3	-8/0.16	2.7/1.86	28/640

O sinal à saída do *front-end* terá uma potência mínima de:

$$P = -95 \text{ dBm} + 28 \text{ dB} = -67 \text{ dBm} \quad (4.2)$$

E uma amplitude mínima de:

$$V_p = \sqrt{2P \cdot R} \approx 141 \mu V \quad (4.3)$$

Em relação ao ruído, admitindo o pior caso de temperatura ( $360 \text{ K} = 87^\circ \text{C}$ ) e considerando a largura de banda do sinal  $2B = 500 \text{ KHz}$ , à entrada do recetor temos um nível de ruído térmico de:

$$P_n = kTB = 2.5 \text{ fW} = -116 \text{ dBm} \quad (4.4)$$

Assim, no terminal da antena, o SNR é de:

$$\text{SNR} = -95 \text{ dBm} - (-116 \text{ dBm}) = 21 \text{ dB} \quad (4.5)$$

Pela Tabela 4.2 verificou-se que o fator de ruído acumulada pela cadeia é de aproximadamente 3 dB (2.7 dB).

Resulta então que o SNR no fim do *front-end* é de ( $21 \text{ dB} - 3 \text{ dB} = 18 \text{ dB}$ ) e a potência do ruído é de:

$$P = -95 \text{ dBm} + 28 \text{ dB} - 18 \text{ dB} = -85 \text{ dBm} = 2.5 \text{ pW} \quad (4.6)$$

A fonte de tensão de ruído em função da largura de banda é de:

$$\langle |V_n^2| \rangle = 4 RkTB = 4 RP_n = 5 \times 10^{-10} V^2 \quad (4.7)$$

$$\langle |V_n^2| \rangle / B = 1 \times 10^{-15} V^2 / \text{Hz} \quad (4.8)$$

$$\langle |V_n| \rangle / \sqrt{B} = 32 \text{ nV} / \sqrt{\text{Hz}} \quad (4.9)$$

E para toda a banda (500 kHz) corresponde a uma fonte de ruído de:

$$\langle |V_n| \rangle = 22.4 \mu V_{rms} \quad (4.10)$$

Em conclusão, teremos os seguintes dados:

- Potência do sinal: -67 dBm
- Potência do ruído: -85 dBm
- Amplitude do sinal: 141  $\mu V$
- Amplitude do ruído: 22.4  $\mu V_{rms}$
- SNR (Relação sinal/ruído): 18 dB

Com estes valores de sinal e de ruído dever-se-á atingir um *Bit Error Rate (BER)* de  $10^{-6}$  ou menor, estabelecida pela norma EN12253 já referida anteriormente na secção 2.1.

Para a transmissão em questão de *Uplink* é usada a modulação BPSK e a codificação NRZI. Este conjunto pode ser visto como uma transmissão em modulação *Differential Phase Shift Keying (DPSK)*. Define-se como um PSK diferencial, quando o símbolo desmodulado depende do símbolo anterior. Assim, se um dado símbolo for mal desmodulado, o erro propaga-se para o símbolo seguinte. A probabilidade de erro aumenta face ao vulgar PSK. A Figura 4.4 mostra a comparação entre o vulgar PSK e o DPSK. Nesta comparação, no que concede ao PSK, só é aproximadamente válido para SNR elevado.

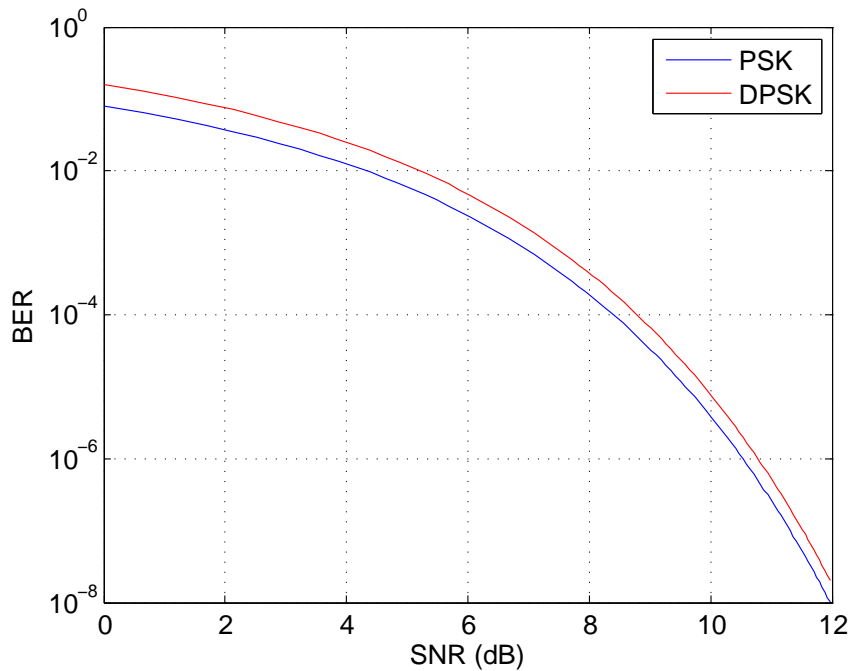


Figura 4.4: Comparação da probabilidade de erro de símbolos (BER) entre PSK e DPSK em função do SNR.

Observa-se então que para uma probabilidade de erro de  $10^{-6}$ , o SNR mínimo do DPSK é de 10.8 dB, ligeiramente superior ao SNR do PSK que é de 10.5 dB.

No entanto, a norma ETSI EN 300 674-1 V1.2.1 (2004-08) indica uma potência típica de -104 dBm para o terminal da antena. Isto significa uma potência reduzida em 9 dB (de -95 dBm para -104 dBm), passando o SNR de 18 dB para 9 dB, insuficiente para o atual sistema de modo a cumprir a meta de  $10^{-6}$  de BER. A norma ETSI EN 300 674-1 V1.2.1 (2004-08) também especifica a potência máxima no terminal da antena, -50 dBm (que corresponde à saída do *front-end* a uma amplitude de 51 mV), para prevenir um eventual mau funcionamento com sinais muito fortes.

### 4.3 Andar de Amplificação, Filtragem e Conversão AD

A implementação deste andar tem como principais objetivos:

- Amplificar o sinal recebido de forma a garantir que mesmo a menor amplitude prevista à entrada seja aproveitada ao máximo da gama de leitura de tensão da ADC.
- Aproveitar o andar de amplificação para filtrar o sinal recebido, construindo assim um filtro passa-banda ativo.
- Utilizar uma ADC para converter os sinais analógicos em sinais digitais para que de seguida sejam tratados digitalmente pela FPGA.

O diagrama de blocos completo deste andar é visível na Figura 4.5.

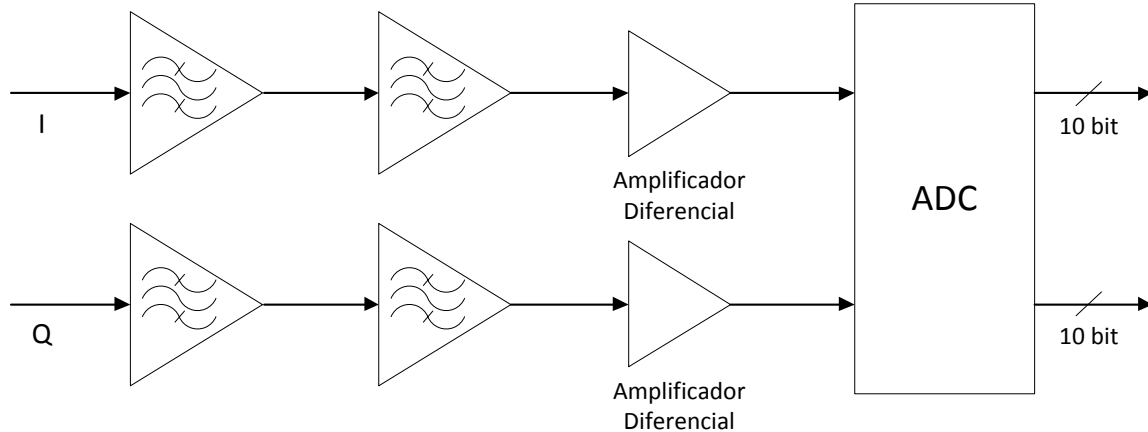


Figura 4.5: Diagrama de blocos do andar de amplificação, filtragem e conversão analógico-digital.

Pela figura observa-se que os ramos do sinal em fase (I) e em quadratura (Q) são idênticos, por essa razão o estudo vai visar apenas um dos ramos. Na fase de desenho da placa são depois constituídos os dois ramos de forma análoga.

A cadeia é assim constituída por dois andares de amplificação e filtragem muito semelhantes e um amplificador diferencial antes da ADC. Este amplificador diferencial tem como objetivo converter a entrada em duas saídas constituindo um sinal diferencial. Esta condição é um dos requisitos necessários para a ADC, já que esta necessita de sinais diferenciais nas suas entradas.

### 4.3.1 Amplificador e filtro passa-banda

O esquema elétrico do amplificador e filtro passa-banda é representado na Figura 4.6.

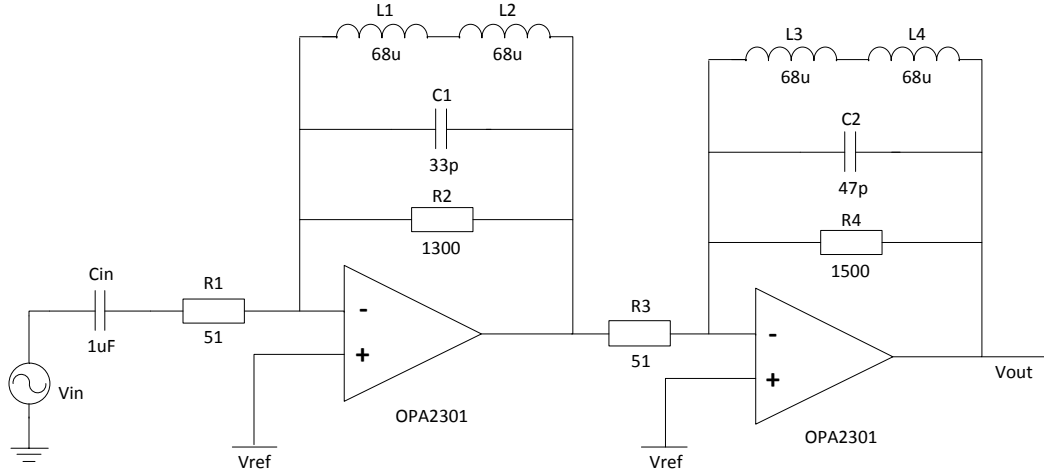


Figura 4.6: Esquema elétrico do circuito do amplificador e filtro passa-banda.

O Amplificador Operacional (AmpOp) usado (OPA2301) foi escolhido por satisfazer alguns requisitos como:

- Produto ganho por largura de banda alto;
- Baixo ruído:  $3 \text{ nV}/\sqrt{\text{Hz}}$ ;
- Boa gama de amplitudes de saída:  $(V^+) - 100\text{mV}$  até  $(V^-) + 100\text{mV}$ ;
- Compatível com alimentação a 3.3V;
- Dois AmpOps num só circuito integrado.

A função de transferência em cada AmpOp é dada por (para o AmpOp da esquerda):

$$H = -\frac{Z_{eq}}{R1} \quad (4.11)$$

Em que:

$$Z_{eq} = R2 \parallel \frac{1}{sC1} \parallel s(L1 + L2) \quad (4.12)$$

Isto resulta num ganho em tensão de:

$$G_V = -\frac{R2}{R1} \quad (4.13)$$

O cálculo da largura de banda do filtro passa-banda resultante é dado por:

$$B = \frac{1}{2\pi R2 C1} \quad (4.14)$$

E a frequência central do filtro é calculada através de:

$$f_R = \frac{1}{2\pi\sqrt{(L1 + L2)C1}} \quad (4.15)$$

São usadas duas indutâncias por cada AmpOp, pois foi esta a forma mais prática de conseguir os valores pretendidos de frequência central e de largura de banda.

Para o primeiro AmpOp haverá um ganho em tensão de:

$$G_V = -25.5V/V = 28dB \quad (4.16)$$

E para o segundo será de:

$$G_V = -29.4V/V = 29.4dB \quad (4.17)$$

No conjunto dos dois:

$$G_V = \frac{V_{out}}{V_{in}} = -25.5 \times -29.4 = 750V/V = 57.4dB \quad (4.18)$$

O condensador  $C_{in}$  tem como fim o desacoplamento das tensões aos seus terminais. Isto permite que o sinal após o condensador seja centrado na tensão de referência  $V_{ref}$  que liga a cada um dos AmpOps. Esta tensão  $V_{ref}$  é sensivelmente metade da amplitude da tensão de alimentação e é gerada pela ADC, como será explicado na secção correspondente. Assim, o sinal fica centrado entre a massa e a tensão de alimentação.

Os valores dos condensadores (C1 e C2) e das indutâncias (L1 a L4) foram obtidas através das relações 4.14 e 4.15 e através de simulação com o auxílio da aplicação OrCAD Capture.

O resultado da simulação comparado à resposta em frequência obtida através de um *Vector Network Analyzer (VNA)* é representado através da Figura 4.7.

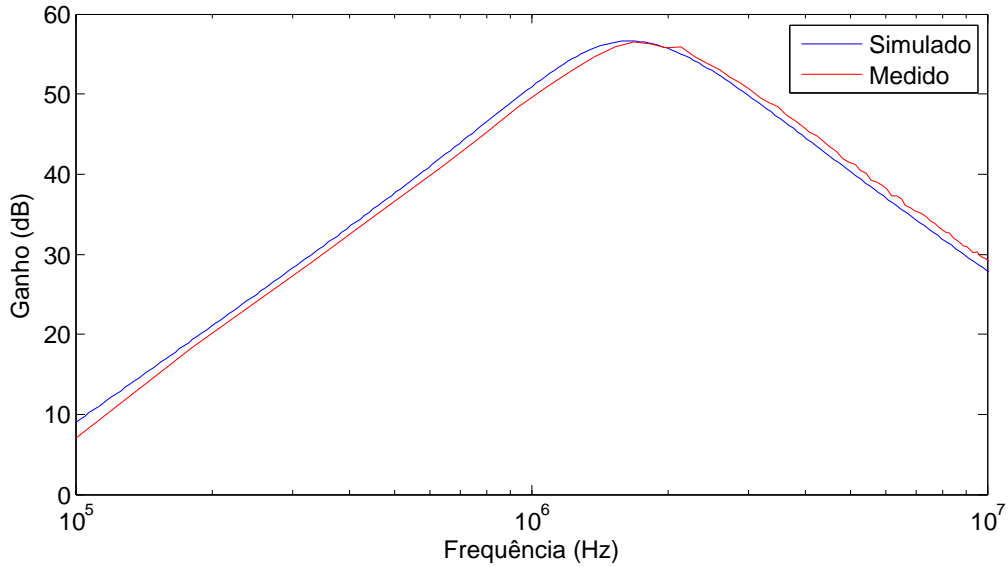


Figura 4.7: Comparação entre a simulação e o resultado retirado através do VNA da resposta em frequência do andar de amplificação e filtragem.



Recorde-se da secção 3.8 que o filtro teria de deixar passar uma largura de banda de 1.25 MHz a 2.25 MHz.

Da Figura 4.7 retiramos as seguintes especificações. Da simulação:

- Ganho máximo: 56.6 dB a 1.65 MHz;
- Corte a 3 dB: 1.17 MHz e 2.36 MHz;
- Largura de banda: 1.19 MHz;
- Decaimento de aproximadamente 40 dB/dec.

Do resultado do VNA:

- Ganho máximo: 56.5 dB a 1.68 MHz;
- Corte a 3 dB: 1.22 MHz e 2.42 MHz;
- Largura de banda: 1.2 MHz;
- Decaimento de aproximadamente 40 dB/dec.

É possível concluir que os resultados estão bastante próximos. A diferença de quase 1 dB do ganho máximo da medição (56.5 dB) com o ganho calculado na equação 4.18 (57.4 dB) está certamente relacionada com a carga de  $50\ \Omega$  imposta pelo porto de saída do VNA na medição. Esta carga é relativamente baixa e pode influenciar o comportamento do amplificador / filtro. Na simulação também foi usada uma carga de  $50\ \Omega$  à saída para que os resultados fossem semelhantes. No entanto, na implementação esta carga deixa de ter efeito, pois o andar seguinte tem uma resistência de entrada de  $510\ \Omega$  que não prejudicará o comportamento deste amplificador / filtro.

O decaimento de 40 dB/dec é o esperado, já que se trata de um filtro de 2ª ordem (dois filtros ativos RLC de primeira ordem em cascata).

Assim, este andar de amplificação e filtragem consegue cumprir os objetivos traçados – amplificar o sinal de entrada aproveitando os AmpOps para constituir também um filtro tendo em conta as duas especificações da norma (sub-portadora a 1.5 MHz ou a 2 MHz).

### 4.3.2 Amplificador diferencial

O esquema elétrico do circuito do amplificador diferencial é representado na Figura 4.8.

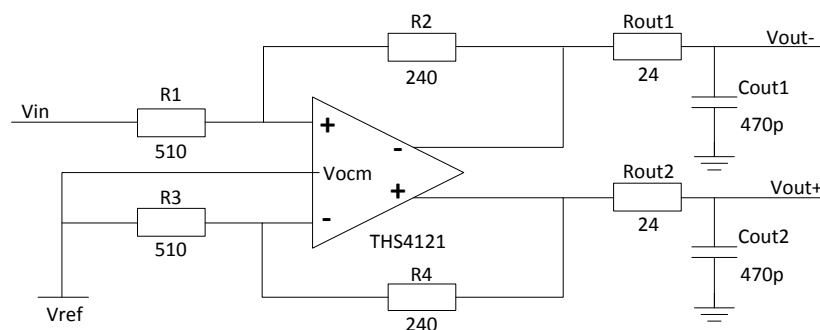


Figura 4.8: Esquema elétrico do circuito do amplificador diferencial.

A montagem indicada na figura tem como objetivo a conversão de um sinal *single-ended* para diferencial. O ganho em tensão do AmpOp THS4121 é dado por:

$$G_V = \frac{R_2}{R_1} = \frac{R_4}{R_3} \quad (4.19)$$

Assim, as resistências têm de ser iguais entre os dois ramos de realimentação do AmpOp ( $R_1 = R_3$  e  $R_2 = R_4$ ).

A saída diferencial do AmpOp é relacionada com uma saída normal *single-ended* da seguinte forma:

$$V_{out} = V_{out+} - V_{out-} \quad (4.20)$$

As resistências  $R_{out1} = R_{out2}$  e os condensadores  $C_{out1} = C_{out2}$  na saída do AmpOp formam um filtro passa-baixo para evitar eventuais perturbações e interferências de altas frequências que prejudiquem a performance da ADC.

É ainda possível verificar pela Figura 4.8 que o ganho / atenuação previsto para o sinal foi de aproximadamente 1/2. A razão para este ganho é explicado pela necessidade de inserir o sinal na gama de leitura de tensão da ADC. Para o caso extremo, quando o sinal é saturado nos AmpOps do andar anterior, a amplitude do sinal será (recordando o ganho no AmpOp diferencial 240/510 e considerando que  $V_{ref}=1.65V$ , metade da tensão de alimentação do circuito):

$$V_p = (3.3V - 100mV - 1.65V) \cdot \frac{240}{510} \approx 0.73V \quad (4.21)$$

### 4.3.3 ADC

A ADC usada neste trabalho foi uma ADC da MAXIM, a MAX1184. O diagrama funcional da mesma pode ser visualizado na Figura 4.9.

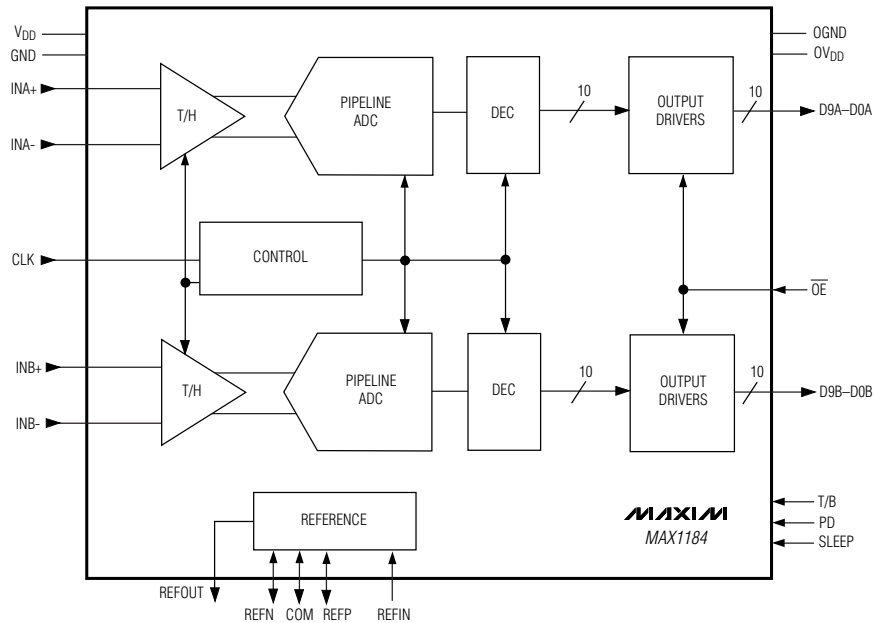


Figura 4.9: Diagrama funcional ADC MAX1184. [MAX06]

Como é possível analisar pela figura, esta ADC é do tipo *pipelined*. Contém algumas características importantes para o trabalho. As mais relevantes a destacar são as seguintes:

- Conversão analógico para digital de 10 *bits*;
- Converte dois sinais em paralelo, ou seja, os instantes de conversão não são *multiplexados* no tempo entre as duas entradas;
- Saídas paralelas com transmissão de 10 *bits* de cada sinal;
- ADC *pipelined* de 9 estágios;
- Uma só alimentação para as partes analógica e digital compatível com 3.3 V;
- Entradas analógicas diferenciais com gama máxima de 1 Vpp.
- Gerador interno de tensão de referência de 2.048 V.
- Frequência máxima de amostragem de 20 Msps;
- Aproximadamente 60 dB de SNR com a frequência do sinal de entrada abaixo de 10 MHz;
- Latência entre a entrada e a saída de 5 ciclos de relógio.

#### 4.3.3.1 Seleção da gama de amplitudes de tensão da entrada da ADC

A presente ADC tem 2 tipos de seleção para a gama de tensões a ler nas suas entradas. São elas:

- Modo de referência interno. Este modo utiliza a fonte de tensão de referência interna da ADC para produzir a amplitude mínima e máxima da gama a ler.
- Modo de referência externo. Este modo utiliza uma fonte de tensão externa disponibilizada pelo utilizador para garantir uma gama de tensões diferente à resultante do modo anterior. Normalmente é utilizado este modo para gamas muito pequenas, necessitando de tensões de referência muito precisas.

Para este trabalho foi utilizado o primeiro método. Estando o sinal de entrada nesta fase bastante amplificado, pode-se usar a referência disponibilizada internamente pela ADC. As amplitudes mínima e máxima são dadas por:

$$V_{DD}/2 - V_{REFIN}/4 < V_{IN} < V_{DD}/2 + V_{REFIN}/4 \quad (4.22)$$

REFIN é a tensão de referência interna de 2.048V.

Assim, no presente projeto, com  $V_{DD} = 3.3V$  temos que a tensão nas entradas pode variar entre:

$$1.138 \text{ V} < V_{IN} < 2.162 \text{ V} \quad (4.23)$$

Ora, vimos pelo resultado 5.2 que a amplitude máxima do sinal de entrada é de aproximadamente 0.73 V (que em amplitude diferencial passa para metade). Assim, a tensão de entrada do sinal terá uma gama de:

$$1.285 V < V_{\text{signal}} < 2.015 V \quad (4.24)$$

Este intervalo é, portanto, próximo da gama de amplitude máxima permitida pela ADC. Utilizado o modo de referência externo podia-se ter diminuído a referência da ADC de modo a ficar mais próximo ainda da amplitude máxima do sinal. No entanto, chegou-se à conclusão que não era necessário melhor aproveitamento da amplitude do sinal de entrada. Usando o modo de referência interno, o sistema fica também mais simples, uma das preocupações na elaboração do projeto.

Para a amplitude mínima prevista à saída do *front-end*, 141  $\mu V$ , teremos a seguinte amplitude do sinal à entrada da ADC:

$$V_p = 141 \mu V \times 750 \times (240/510) = 0.05 V \quad (4.25)$$

Por ser diferencial, passa para metade a amplitude anterior, e a gama resultante será:

$$1.625 V < V_{\text{signal}} < 1.675 V \quad (4.26)$$

Este intervalo em número de níveis da ADC será:

$$0.05 V \times 2^{10} / 1.024 V = 50 \quad (4.27)$$

Esta amplitude mais baixa foi simulada de forma a provar que era suficiente para concretizar a desmodulação do sinal.

É aproveitada ainda a saída COM (visível na Figura 4.9) como tensão de referência nos AmpOps representada nas secções anteriores por  $V_{\text{ref}}$ . Esta saída é também gerada automaticamente de forma interna pela ADC como metade da tensão de alimentação  $V_{DD}$  e é exatamente o objetivo desta porta servir de referência para ser possível outros dispositivos centrarem o sinal de entrada. Assim, é garantido que a tensão DC do sinal é igual em toda a placa, desde do início da amplificação até à ADC.

#### 4.3.3.2 Seleção das entradas digitais de controlo da ADC

Existem 4 entradas digitais de controlo da ADC (que podem ser observadas na Figura 4.9):  $\overline{OE}$ , T/B, PD e SLEEP.

- $\overline{OE}$  *Output Enable Input*. Esta entrada ativa ou desativa a saída dos dados da ADC.
- T/B. Esta entrada seleciona o formato dos dados da saída. Em complemento para 2 ou unipolar (isto é, de forma linear e sem sinal).
- PD *Power-Down Input*. Esta entrada ativa ou desativa o modo de operação de baixa potência da ADC.
- SLEEP. Seleciona ou não o modo de suspensão da ADC, isto é, desativa as duas ADCs mas deixa os mecanismos das tensões de referência ativos.

Estas entradas de controlo estão ligadas à placa de desenvolvimento da FPGA, tal como as saídas da ADC e a entrada do relógio (CLK). O relógio usado é portanto o relógio gerado internamente na FPGA de 16 MHz que também será posteriormente usado na desmodulação, tal como também foi usado no capítulo da simulação do desmodulador.

O esquema elétrico e o *layout* PCB da placa de amplificação, filtragem e ADC podem ser consultados no Apêndice A.

## 4.4 Implementação do Desmodulador na FPGA

### 4.4.1 Esquema geral do desmodulador

O esquema da Figura 4.10 ilustra o desmodulador implementado na FPGA.

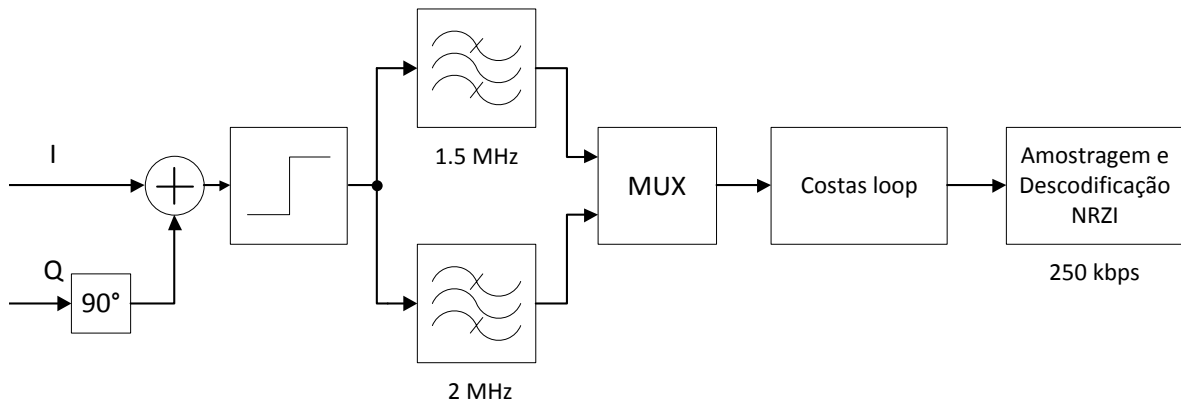


Figura 4.10: Esquema do desmodulador na FPGA.

Este desmodulador está preparado para desmodular sinais recebidos com sub-portadora a 1.5 MHz ou a 2 MHz. Como referido na secção 2.1, num sistema real a escolha da frequência de sub-portadora que o OBU deve usar é efetuada pelo RSU no processo de *Downlink*. Assim, quando o RSU está pronto a receber os sinais na comunicação de *Uplink*, este tem presente qual a frequência da sub-portadora que o OBU transmitiu e seleciona as características do desmodulador consoante essa frequência. Estas características a selecionar são nomeadamente a escolha do filtro passa-banda (efetuado através do *multiplexer* representado na Figura 4.10) e a frequência do oscilador local do *Costas loop*.

No desmodulador implementado, como se trata de um sistema de teste, ou seja, e é independente da outra parte do RSU – o emissor – a escolha é efetuada através de um DIP Switch da placa de desenvolvimento da FPGA.

Nas próximas sub-secções serão descritos os blocos implementados na FPGA.

#### 4.4.2 Atraso de 90° no sinal em quadratura

Como já referido anteriormente na simulação do desmodulador, a frequência de amostragem é de 16 MHz. O cálculo do número de amostras a atrasar de forma a provocar um atraso de 90° no sinal em quadratura de entrada é dado por:

$$\text{amostras}_{90} = \frac{f_{CLK}}{f_{sinal}} \times \frac{\pi}{2} \quad (4.28)$$

Assim, para 2 MHz teremos 2 amostras e para 1.5 MHz teremos aproximadamente 2.67 amostras. Por estarmos a uma frequência de amostragem relativamente baixa, não existe precisão suficiente para que o número de amostras para as duas opções de frequências do sinal de entrada sejam um número inteiro. Para contornar este problema, haviam duas possíveis soluções:

- Aumentar a frequência de amostragem até conseguir um número inteiro para as duas opções (comutando depois entre os 2 números através de um *multiplexer*). A primeira frequência possível era de 24 MHz. No entanto, seria necessário usar uma ADC preparada para uma frequência de amostragem superior, já que a apresentada está limitada à frequência de 20 MHz. Esta opção levaria também ao projeto de filtros com maior precisão aumentando assim o seu número de ordem correndo o risco de saturar os recursos da FPGA.
- Implementar o sistema com duas frequências de amostragem distintas, uma para cada frequência do sinal de entrada. No entanto, isto levaria a uma abordagem em que seria necessário projetar filtros para cada uma das frequências de amostragem, que tornaria a implementação do sistema bem mais complexa. Não houve oportunidade de efetuar esta opção durante a execução deste trabalho, pelo que é uma das recomendações para o trabalho futuro de evolução.

Por nenhuma das soluções anteriores ter uma implementação aparentemente simples, foi usado o mesmo número de amostras de atraso para as duas opções de frequência do sinal. O número usado foi o de 2 amostras. Para a frequência de 2 MHz significa que o atraso é exato em relação aos cálculos estabelecidos, mas para a frequência de 1.5 MHz provocará um erro de fase de aproximadamente  $0.39 \text{ rad} = 22.35^\circ$ <sup>1</sup>.

#### 4.4.3 Saturação do sinal

Foi visto anteriormente que consoante a amplitude que chega do *front-end*, o número de níveis representados pela ADC varia.

O número de níveis varia entre 50 (resultado 4.27) e 730 (resultado 4.24). Também foi visto na sub-seção 3.3.2 na relação 3.32 que o ganho do *Costas loop* é dependente da amplitude do sinal de entrada.

De forma a contornar este inconveniente, antes de iniciar a desmodulação, o sinal é saturado a um nível equivalente ao valor mínimo de amplitude previsto, ou seja, aos 50 níveis. Estes 50 níveis equivalem a um intervalo em complemento para 2 de -25 a 24. Para usar um número de potência de 2, determinou-se que o sinal era saturado no nível inferior a -16 e no nível superior a 15, transformando o sinal num sinal de apenas 5 *bits*.

---

<sup>1</sup>Não foram efetuadas simulações para conhecer aparentemente quais os efeitos no sistema do presente erro de fase.

#### 4.4.4 Filtros passa-banda

Foram construídos dois filtros passa-banda através da ferramenta da Xilinx, o *Core* “FIR Compiler”. Cada um dos filtros foi projeto para cada uma das opções de frequência do sinal (1.5 MHz e 2 MHz). Estes filtros foram projetados com base na simulação já efetuada. A grande diferença entre o filtro da simulação e os filtros projetados passa pelo facto dos filtros implementados terem coeficientes inteiros. Este pormenor deve-se à razão de que no projeto implementado na FPGA não ter sido usada vírgula flutuante. A abordagem foi sempre com valores inteiros (em complemento para 2).

As respostas em frequência dos dois filtros implementados podem ser visualizados na Figura 4.11.

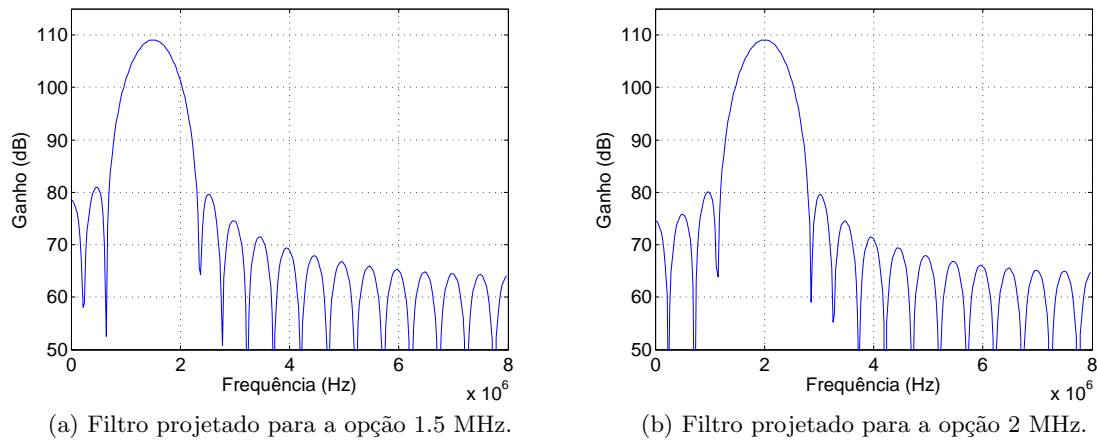


Figura 4.11: Resposta em frequência dos dois filtros passa-banda implementados na FPGA.

Estes filtros contêm um ganho máximo na largura de banda desejada (de 1.25 MHz a 1.75 MHz e de 1.75 MHz a 2.25 MHz) de aproximadamente 109 dB. O corte a 3 dB é situado a aproximadamente 1.18 MHz, 1.83 MHz e 1.68 MHz, 2.33 MHz, respetivamente. A ordem dos filtros é de 30.

De forma a equivaler estes filtros a filtros comuns, ou seja, atenuando a gama de frequências indesejáveis mantendo a amplitude da largura de banda de interesse intacta (0 dB), à saída dos filtros é aplicado um atenuador do valor do ganho máximo.

A forma como isso é efetuado é dividindo a amplitude do sinal pelo valor desse ganho, que na FPGA se resume à realização de um *shift-right*. O cálculo do número de *bits* menos significativos a desprezar é dado por:

$$\log_2 \left( 10^{109/20} \right) \approx 18 \quad (4.29)$$

#### 4.4.5 Síntese do relógio de 16 MHz

Apesar da placa de desenvolvimento usada conter alguns geradores de relógio, nenhum destes tinha a frequência pretendida para este trabalho, 16 MHz. No entanto, as ferramentas de desenvolvimento da FPGA têm à disposição do utilizador o *Core Digital Clock Manager* (DCM).

Este *Core* contém inúmeras funções tornando a sua análise de alguma forma complexa. Neste trabalho, usou-se uma das funções disponíveis – a capacidade de síntese de uma frequência de relógio a partir de um relógio externo inserido à entrada da FPGA. O relógio usado à entrada da FPGA, disponível portanto na placa de desenvolvimento, foi o relógio de 100 MHz.

De forma a obter a frequência de 16 MHz, foram usados dois blocos DCM. O primeiro para transformar a frequência de 64 MHz a partir do relógio externo de 100 MHz através de um multiplicador de frequência de 16 e de um divisor de frequência de 25:

$$100 \text{ MHz} \times (16/25) = 64 \text{ MHz} \quad (4.30)$$

E de seguida outro DCM para transformar a frequência de 16 MHz a partir do relógio de 64 MHz, através de um divisor de frequência de fator 4.

Foi necessário usar estes 2 blocos e não apenas um, pois no modo de gerar uma frequência por multiplicação e divisão (usado no primeiro bloco), este não suporta relógios à saída com frequências menores a 32 MHz.

#### 4.4.6 *Costas loop*

Como foi estudado nos capítulos anteriores, o *Costas loop* representa o bloco que efetivamente desmodula o sinal transmitido. Enquanto é recuperada a sub-portadora do sinal de 1.5 MHz ou 2 MHz, esta é ao mesmo tempo multiplicada pelo sinal. Dentro da malha ainda, o sinal é filtrado de forma a que na saída seja apresentado o sinal desmodulado limpo a 250 kbps.

O diagrama de blocos da Figura 4.12 representa toda a malha do *Costas loop* implementada na FPGA incluindo as funções nela usadas.

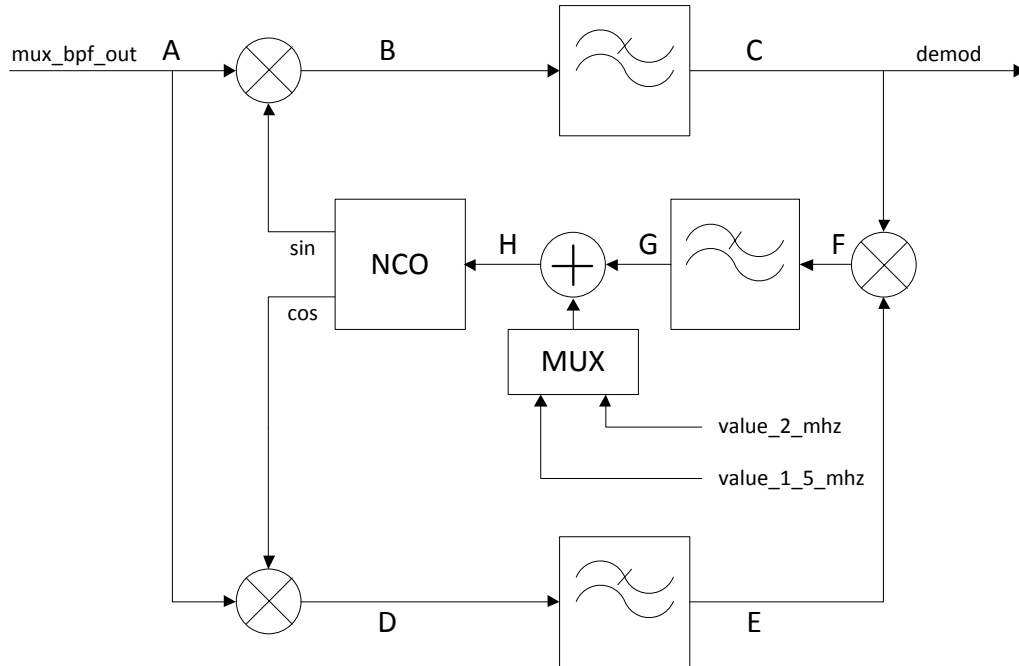


Figura 4.12: Diagrama de blocos do *Costas loop* implementado na FPGA.



#### 4.4.6.1 Numerically-controlled oscillator (NCO)

O *Numerically-Controlled Oscillator (NCO)* é um gerador de sinais digitais que cria uma representação de uma forma de onda discreta no tempo, geralmente sinusoidal. É cada vez mais utilizado em sistemas de telecomunicações digitais, nomeadamente na substituição de VCOs geralmente utilizados em sistemas analógicos.

O NCO é composto por dois blocos principais (como é possível observar na Figura 4.13):

- Um Acumulador de Fases: soma à saída do bloco o valor “Phase Increment”, sincronamente;
- Um Conversor de Fase para Amplitude: usa o valor de saída do acumulador (fase da onda) como endereço de uma *Lookup Table (LUT)* que contém a amplitude correspondente.

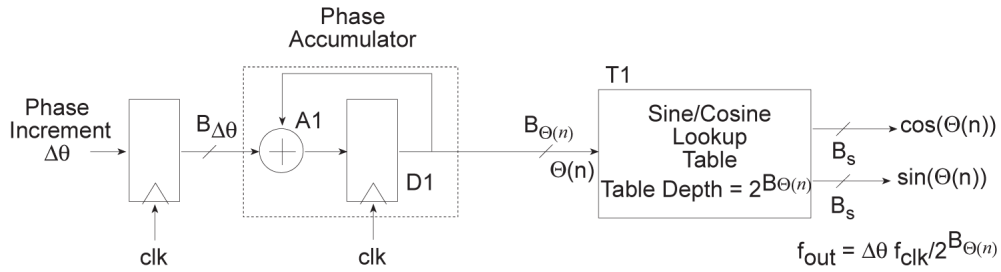


Figura 4.13: Sistema de blocos de um NCO.

Da análise da Figura 4.13, é útil referir o significado das variáveis mencionadas:

- $\Delta\theta$ , é o valor a incrementar no acumulador de fases. Está relacionado com o número total de fases possíveis, e com a frequência que se pretende obter à saída;
- $B_{\Delta\theta}$ , é o número de *bits* usados para representar o valor de incremento de fase;
- $B_{\Theta(n)}$ , é o número de *bits* usados para representar o valor de fase a ser aplicado à LUT. Este é, como é lógico, igual ao número de *bits*  $B_{\Delta\theta}$ ;
- $B_s$ , é o número de *bits* usados para representar o valor da amplitude da forma de onda da respetiva fase.

Portanto,  $B_{\Theta(n)}$  e  $B_s$  estão relacionados com a dimensão da LUT ( $B_{\Theta(n)} \times B_s$ ).

Para facilitar o projeto da LUT foi utilizado o *Core* das ferramentas da Xilinx, o *Direct Digital Synthesizer (DDS)*. Este *Core* cria toda a lógica necessária para a construção de um NCO, sendo apenas necessário indicar alguns valores importantes para o projeto deste como são o caso dos valores  $B_{\Theta(n)}$  e  $B_s$ .

Para representar as duas formas de onda pretendidas (1.5 MHz e 2 MHz) foi necessário calcular os dois valores de incremento de fase correspondentes  $\Delta\theta$  para aplicar à entrada do NCO (na Figura 4.12 – “value\_2\_mhz” e “value\_1.5\_mhz”).

A frequência da forma de onda de saída do NCO é dada por:

$$f_{out} = \frac{f_{clk} \times \Delta\theta}{2^{B_{\Theta(n)}}} \quad (4.31)$$

E assim, os dois valores de incremento de fase  $\Delta\theta$  são dados por:

$$\Delta\theta = \frac{f_{\text{out}} \times 2^{B_{\Theta(n)}}}{f_{\text{clk}}} \quad (4.32)$$

Ora, com a frequência do relógio de amostragem de 16 MHz e usando 16 *bits* para representar o valor de fase a ser aplicado à LUT ( $B_{\Theta(n)}$ ), os valores a serem aplicados ao NCO são:

- “value\_2\_mhz”  $\rightarrow$  8192;
- “value\_1.5\_mhz”  $\rightarrow$  6144.

Um destes valores, conforme o que for selecionado no *multiplexer*, é depois adicionado à amplitude do erro resultante da realimentação (G na Figura 4.12).

O facto de o NCO ter duas saídas (uma *sin* e outra *cos*), e por o *cos* ser uma senoide com atraso de 90° em relação ao *sin*, é aplicada a saída *sin* ao ramo correspondente ao sinal em fase e a saída *cos* ao ramo em quadratura.

#### 4.4.6.2 Filtro passa-baixo (C e E)

Este filtro é praticamente igual ao usado anteriormente na simulação. Assim, nas ferramentas da Xilinx, nomeadamente no *Core* “FIR Compiler”, foi projetado o filtro com base no filtro apresentado na simulação. Para os dois filtros a usar, bastou projetar uma única vez, já que seguidamente foi instanciado o mesmo filtro para os dois pontos necessários.

A resposta em frequência deste filtro é praticamente igual ao da simulação. A diferença passa por, mais uma vez, o seu ganho máximo na banda de passagem ser muito superior a 0 dB. Novamente, isto é devido ao filtro conter coeficientes de números inteiros em vez de usar coeficientes racionais, no âmbito de facilitar a sua implementação na FPGA.

A Figura 4.14 representa a resposta em frequência do filtro implementado.

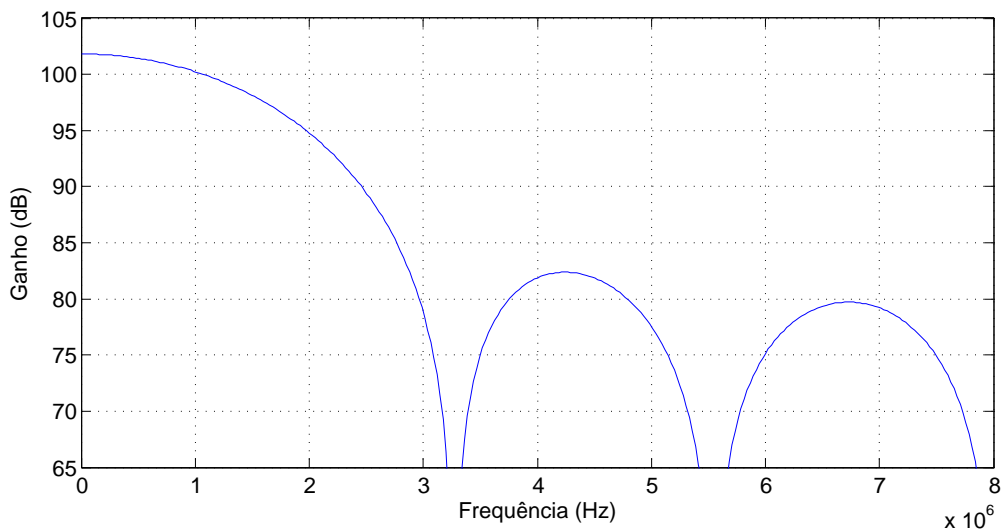


Figura 4.14: Resposta em frequência do filtro passa-baixo usado em C e E do *Costas loop*.

Este filtro contém um ganho máximo em DC de aproximadamente 102 dB. O corte de 3 dB é situado a aproximadamente 1.36 MHz. A ordem do filtro é de apenas 5. Às frequências de 3 MHz e 4 MHz (as riscas de frequências onde é necessário atenuar) a amplitude é de 79 dB e 82 dB respetivamente. Ou seja, existe um corte de 23 dB e 20 dB respetivamente.

Tal como no filtro passa-banda apresentado anteriormente, à saída deste filtro é aplicado um *shift-right* para atenuar o ganho aplicado ao sinal em DC, de forma a que o sinal fique com uma amplitude aproximadamente igual à que tinha à entrada do filtro.

O número de *bits* menos significativos a desprezar é dado por:

$$\log_2 \left( 10^{102/20} \right) \approx 17 \quad (4.33)$$

#### 4.4.6.3 Filtro passa-baixo do erro (G)

Como referido no capítulo anterior, este filtro tem como propósito a estabilização da amplitude do erro que chega à entrada do NCO. Este filtro é projetado como o melhor partido entre dois fatores:

- Se for muito estreito pode tornar o sistema de correção demasiado lento, pois atenua em demasia correções bruscas que podem ser necessárias de fazer;
- Por outro lado, se for bastante largo deixa passar qualquer pico de frequências, estando menos imune à ocorrência de oscilações.

Basicamente, este filtro ajuda a atenuar o ruído e a risca ao dobro da frequência da sub-portadora não atenuados na totalidade pelos filtros anteriores, contribuindo assim para a estabilização do sistema. Mas, como referido nos fatores anteriores, é preciso obter o melhor partido entre as várias hipóteses até chegar à melhor resposta possível do sistema.

Mais uma vez, através das simulações projetadas no capítulo 3, foi possível construir o filtro passa-baixo representado na Figura 4.15.

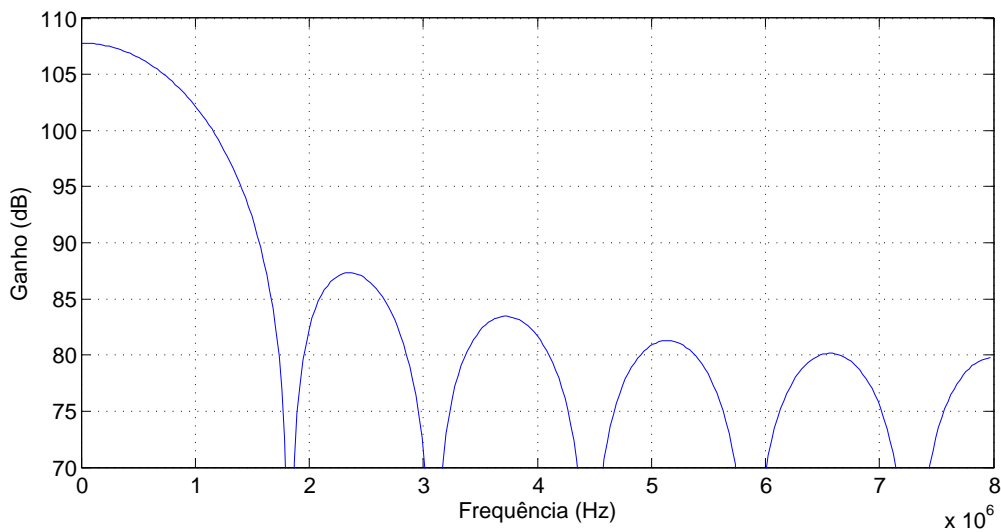


Figura 4.15: Resposta em frequência do filtro passa-baixo usado em G do *Costas loop*.

Este filtro contém um ganho máximo em DC de aproximadamente 108 dB. O corte de 3 dB é situado a aproximadamente 750 KHz, cerca de metade do filtro passa-baixo anterior. A ordem do filtro é de agora 10, o dobro do filtro passa-baixo anterior.

Tal como indicado nos filtros anteriormente analisados, à saída deste filtro também é aplicado um *shift-right* para atenuar o ganho aplicado no sinal em DC, de forma a que o sinal fique com uma amplitude aproximadamente igual à que tinha à entrada do filtro.

O número de *bits* menos significativos a desprezar é dado por:

$$\log_2 \left( 10^{108/20} \right) \approx 18 \quad (4.34)$$

#### 4.4.6.4 Controlo do ganho da amplitude do erro à entrada do NCO

De forma a afinar o comportamento da resposta do *Costas loop*, é necessário ajustar a amplitude do erro a aplicar à entrada do NCO, ou seja, o ganho / atenuação a dar.

No sistema implementado, isso é efetuado através da aplicação de um *shift-right* à saída do multiplicador da malha, ou seja em F na Figura 4.12. Portanto, a amplitude do erro é atenuada.

A atenuação foi efetuada até atingir uma relação que tornasse o sistema suficientemente rápido para que as correções ocorressem apenas durante o tempo do preâmbulo mas que ao mesmo tempo não fosse rápido demais ao ponto de colocar o sistema em oscilação. Isto foi concretizado desprezando mais uma vez um determinado número de *bits* menos significativos. O número obtido de *bits* a desprezar foi de 27 *bits*.

#### 4.4.7 Amostragem e Descodificação NRZI

O diagrama de blocos deste andar é representado na Figura 4.16. Esta fase é composta essencialmente pelo mecanismo de recuperação de relógio já descrito anteriormente e por um decodificador de sinais NRZI. De seguida, vão ser descritos cada um destes blocos implementados na FPGA.

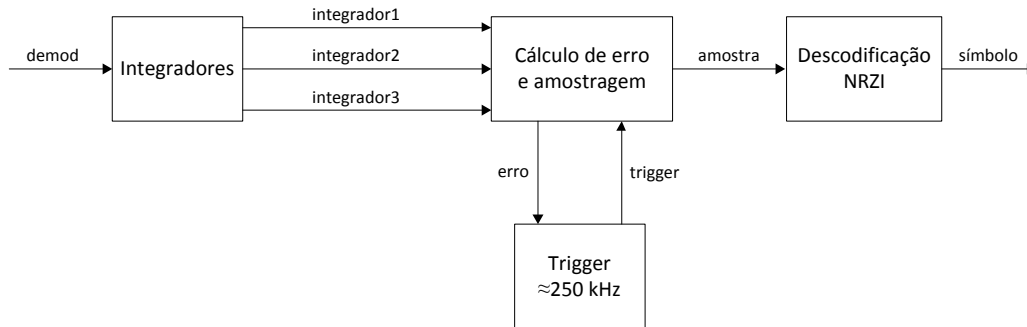


Figura 4.16: Diagrama de blocos da amostragem e decodificação NRZI implementados na FPGA.

##### 4.4.7.1 Integradores

Este bloco tem o propósito de criar três integradores iguais mas atrasados entre si, a partir do sinal desmodulado pelo *Costas loop*. O esquema que visa a representação do funcionamento deste bloco é apresentado na Figura 4.17.

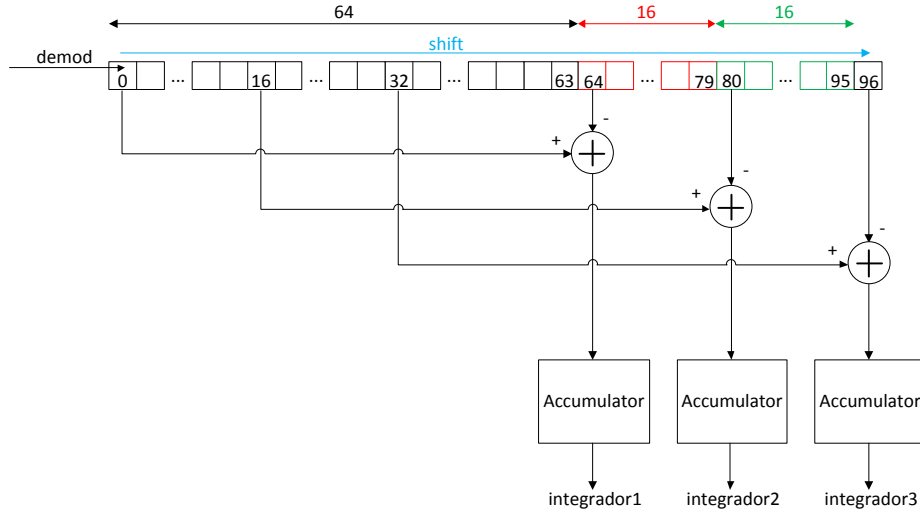


Figura 4.17: Esquema do funcionamento do bloco dos integradores implementado na FPGA.

A integração é efetuada durante um tempo finito. Esse intervalo de tempo é a duração prevista de uma amostra do sinal em banda-base. O número de amostras que representam esse intervalo de tempo é dado por:

$$N = \frac{f_{\text{clk}}}{R} = \frac{16 \text{ MHz}}{250 \text{ kbps}} = 64 \quad (4.35)$$

O atraso provocado para cada um dos outros dois integradores é de  $N/4$  e  $N/2$ , ou seja, 16 e 32 amostras (recorde-se na Figura 3.8 ou 3.30).

Ora, como é verificado pela Figura 4.17 cada integrador é criado pelo auxílio de um acumulador. O acumulador para o *integrador1* adiciona e acumula imediatamente o primeiro índice do *array*, ou seja, sem qualquer atraso. O acumulador para o *integrador2* adiciona e acumula a 16ª posição, e assim sucessivamente.

Para garantir que cada acumulador apenas acumula as última 64 amostras, é subtraída à entrada de cada acumulador a amostra com atraso de 64 ciclos de relógio correspondente, que no caso do acumulador do *integrador1* é a amostra presente no índice 64.

As amostras são movidas no *array* através de um *shift* em cada ciclo de relógio.

#### 4.4.7.2 Cálculo de erro e amostragem

Este bloco produz para a saída “erro”, representada na Figura 4.16, o valor de  $-N/16$  (-4), 0 ou  $N/16$  (4) conforme seja necessário adiantar, manter ou atrasar o *trigger* a 250 kHz.

Isto é determinado por:

- $|(\text{integrador1} - \text{integrador2})| > 0$ , o *trigger* está adiantado e assim é necessário atrasá-lo - o erro será de 4;
- $|(\text{integrador3} - \text{integrador2})| > 0$ , o *trigger* está atrasado e assim é necessário adiantá-lo - o erro será de -4;
- Quando não é verificado nenhuma das opções anteriores, o erro é posto a 0.

Ainda neste bloco, é posto na saída “amostra”, representada também na Figura 4.16, o símbolo de ‘1’ ou ‘0’, conforme o sinal do valor em *integrador2* no instante de amostragem. Se for positivo o símbolo será ‘1’, se for negativo o símbolo será ‘0’.

#### 4.4.7.3 Trigger

Este bloco cria um *trigger* de aproximadamente 250 kHz a partir do relógio do sistema de 16 MHz. Aproximadamente porque este pode variar um pouco, conforme o erro calculado pelo bloco descrito anteriormente.

Na Figura 4.18 é ilustrado um exemplo de funcionamento deste bloco.

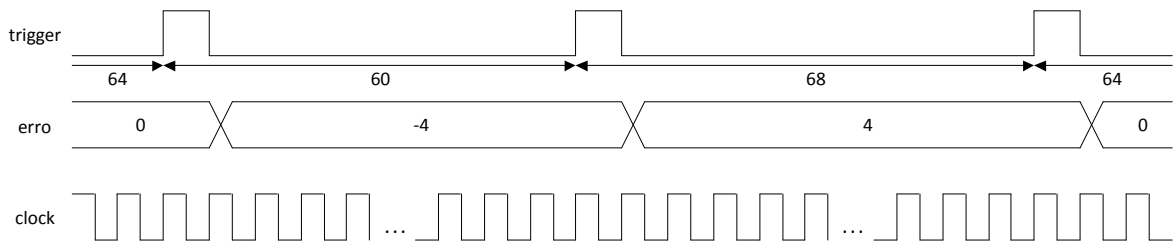


Figura 4.18: Exemplo de funcionamento do bloco “trigger” implementado na FPGA.

Neste caso concreto observa-se que inicialmente o *trigger* é ativado ao fim de 64 ciclos de relógio, os necessários para desmultiplicar o relógio de 16 MHz em 250 kHz. De seguida, a entrada “erro” é comutada para o valor de -4, por isso a próxima ativação ocorrerá 4 ciclos mais cedo, ou seja ao fim de 60 ciclos de relógio. Por fim, é ilustrado ainda o caso em que é necessário atrasar 4 ciclos de relógio, e a ativação ocorre ao fim de 68 ciclos de relógio.

#### 4.4.7.4 Decodificação NRZI

Este bloco tem a função de decodificar os símbolos em NRZI transmitidos em banda-base.

Para o efeito foi implementado uma máquina de estados finitos representada pela Figura 4.19.

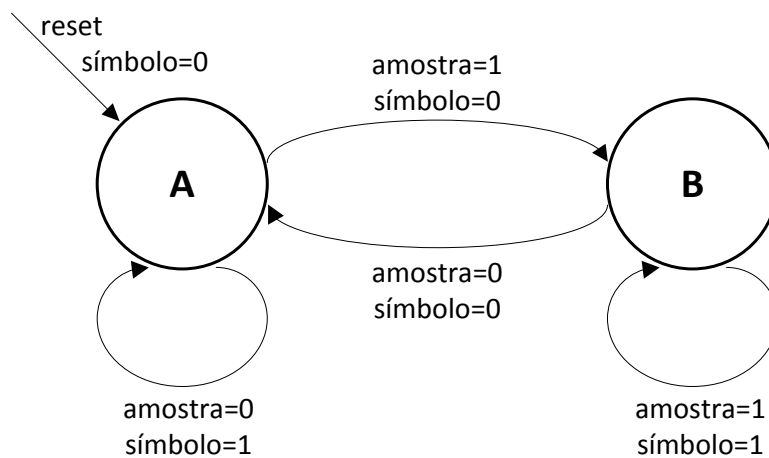


Figura 4.19: Máquina de estados finitos de “Mealy” do bloco de decodificação NRZI implementado na FPGA.

A explicação do funcionamento da máquina de estados finitos de *Mealy* que foi implementada é simples. Cada estado representa a memória do valor da última amostra. O estado A indica que a última amostra foi um 0 e o estado B indica que a última amostra foi um 1.

Assim, se o valor da amostra se mantiver em relação à anterior amostra é porque o símbolo transmitido foi um “1” e a máquina mantém-se no mesmo estado. Se o valor da amostra for diferente em relação à anterior amostra é porque o símbolo transmitido foi um “0” e a máquina transita para o outro estado.

O esquema completo de todo o sistema lógico implementado na FPGA (filtragem, *Costas loop*, amostragem e decodificação) pode ser consultado no Apêndice B.

#### 4.4.8 Placa de desenvolvimento da FPGA

A placa de desenvolvimento usada neste trabalho foi a ML505 para FPGAs Virtex-5 da Xilinx. Esta placa contém algumas características interessantes para o desenvolvimento do desmodulador, entre elas:

- FPGA Xilinx Virtex-5. No modelo da placa disponível para a Universidade, a FPGA é a XC5VLX110T. Esta FPGA contém um vasto array de *slices* de LUTs e de *flip-flops* e ainda 64 *slices* de DSP48Es <sup>2</sup> importantes para o uso de filtros FIR.
- Vasta interface de pinos que permitem ligações paralelas a placas de expansão (neste caso à placa implementada neste trabalho para conversão do sinal de analógico para digital).
- Diversos mecanismos de teste disponíveis: LEDs, DIP Switches, interface JTAG (usada para retirar resultados, através do ChipScope, e para programar a FPGA), etc...

No entanto, lembrando que o interesse do desenvolvimento deste recetor para comunicações DSRC através de técnicas de SDR era principalmente a procura de simplicidade e redução dos custos do sistema, não era obrigatório o uso desta placa com esta FPGA tão avançada. Esta placa foi usada somente por estar disponível no Instituto de Telecomunicações e, por sua vez, conter recursos em excesso, facilitando o estudo em questão.

O esquema da placa pode ser visualizada no Apêndice C onde contém indicações às interfaces e aos dispositivos usados.

---

<sup>2</sup>Cada DSP48E disponibiliza um multiplicador 25 x 18, um somador e um acumulador.





## Capítulo 5

# Resultados

### 5.1 Introdução

Neste capítulo serão constituídas algumas montagens com esquemas de testes que visam a avaliação e validação do sistema implementado.

No primeiro teste, inserindo um sinal de entrada produzido pelo MATLAB, o objetivo é analisar o comportamento do sinal ao longo de todos os pontos do sistema através do ChipScope. Os resultados retirados são confrontados com base nas simulações efetuadas no capítulos 3.

O próximo teste aplicado é semelhante ao anterior, mas a observação dos resultados é agora efetuada com o osciloscópio. Não é possível visualizar os pontos do sistema com tanto detalhe, mas em contrapartida, conseguem-se sobrepor vastas gamas de resultados por ponto durante muito tempo. Os dados são retirados da FPGA através duma DAC. Assim, com a persistência de resultados no osciloscópio, o comportamento do sistema é avaliado para uma grande variedade de casos possíveis.

Por fim, a montagem do terceiro teste tem como base a usada no segundo (osciloscópio e DAC), mas é adicionado ruído ao sinal de entrada de tal forma que a relação SNR na banda de interesse seja equivalente a 18 dB. Este é o SNR previsto para uma situação de utilização real do sistema implementado.

### 5.2 Testes de avaliação e validação através do ChipScope

Para que fosse possível avaliar o funcionamento do sistema implementado, foi constituído inicialmente uma primeira montagem. A montagem usada está representado na Figura 5.1.

Este esquema é constituída por três blocos principais: o uso do MATLAB através de um PC, um *Vector Signal Generator (VSG)* e por fim a utilização do sistema implementado (An-  
dar de conversão AD e FPGA). Foi utilizada apenas uma das entradas da placa de conversão AD (a correspondente ao sinal em fase I).

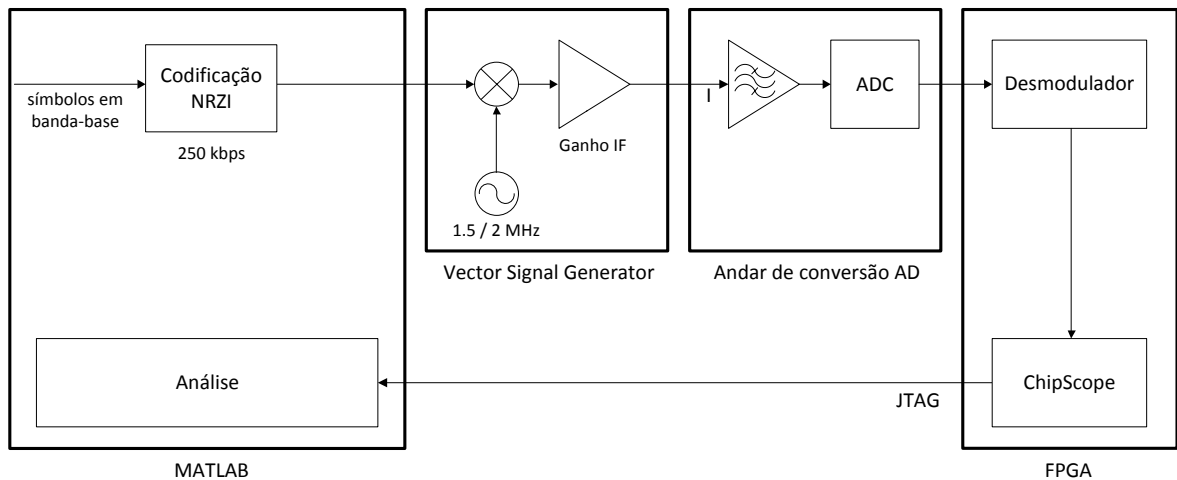


Figura 5.1: Diagrama de blocos da montagem para testes através do ChipScope.

**MATLAB** é utilizado para gerar os símbolos em banda-base a 250 kbps precedidos pelo preâmbulo conforme a norma. Seguidamente os símbolos são codificados em NRZI. A forma como são implementados estes passos no MATLAB é semelhante à usada na simulação do capítulo 3. A matriz de pontos gerada tem uma frequência de amostragem de 16 MHz, igual à usada pelo sistema de desmodulação. Depois de gerada a matriz é transferida para o VSG.

**Vector Signal Generator** é um dispositivo digital com a função de gerar sinais de comunicações de forma avançada em comparação com os tradicionais geradores analógicos. O VSG tem a capacidade de produzir sinais em banda-base com modulações pré-programadas, como por exemplo, QAM, QPSK, FSK, BPSK e OFDM. Depois de produzidos os sinais, este dispositivo tem a capacidade de colocar na saída o sinal analógico equivalente.

Embora durante este teste não seja necessário usar algumas das principais funcionalidades de um VSG, como a produção de sinais em banda-base, este foi útil para transpor o sinal de 250 kbps do MATLAB para 1.5 MHz ou 2 MHz e de seguida transformar em sinal analógico. A conversão de digital para analógica é baseada na frequência de amostragem usada no MATLAB, 16 MHz.

Outra funcionalidade usada no VSG foi a seleção da potência do sinal de saída. Através desta funcionalidade é possível testar o sistema para várias potências de transmissão dentro da gama prevista num sistema real.

**ChipScope** No esquema da Figura 5.1 é possível reparar num bloco adicional dentro da FPGA quando comparado com outros esquemas já descritos no capítulo da implementação. Este bloco usado é denominado por *ChipScope*.

O *ChipScope* é um analisador lógico integrado que permite, através da introdução de lógica adicional ao sistema, a captura em tempo real e o armazenamento de sinais. De seguida os sinais são transmitidos para um PC através da interface de transmissão e programação JTAG.

Através da ferramenta “ChipScope Pro Analyser” da Xilinx é depois possível transferir os dados retirados para o MATLAB e processar uma análise detalhada dos resultados.

### 5.2.1 Saturação e filtro passa-banda digital

Com o objetivo de ser possível indicar o ponto correspondente a cada resultado, a Figura 5.2 relembra o diagrama da implementação desta fase na FPGA.

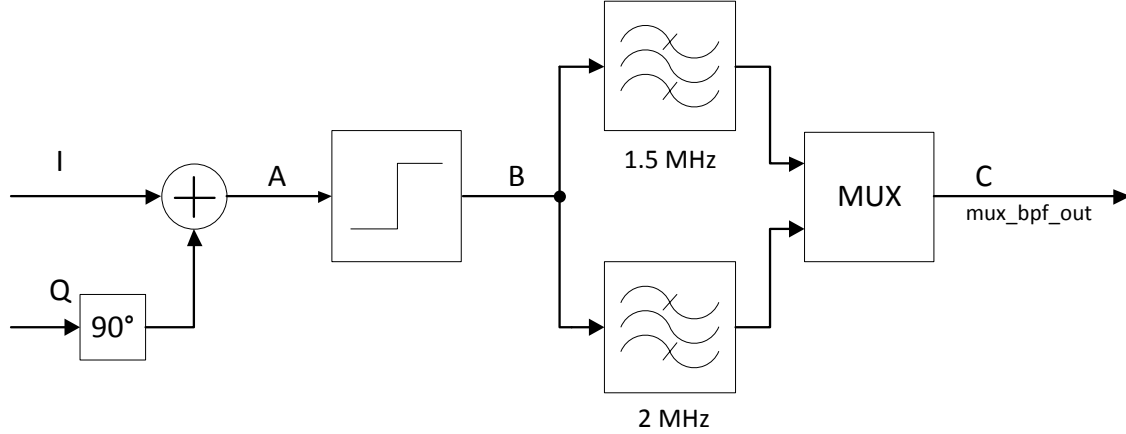


Figura 5.2: Diagrama da fase de saturação e de filtragem na FPGA.

#### 5.2.1.1 Sinal de entrada na FPGA – saída da ADC (A)

À saída do VSG a potência era de -63 dBm. A tensão correspondente a esta potência é:

$$P = \frac{V_p^2}{2R} \Leftrightarrow V_p = \sqrt{2P \cdot R} \approx 224\mu V \quad (5.1)$$

Com o ganho previsto do andar do filtro passa-banda analógico de 57.4 dB, e com a atenuação do andar do amplificador diferencial, a tensão pico a pico à entrada da ADC será de:

$$V_{pp} = 448\mu V \cdot 750 \cdot \frac{240}{510} \approx 0.158V \quad (5.2)$$

Ora, isto corresponde ao número de níveis da ADC de (relembre a tensão pico-a-pico máxima à entrada da ADC de 1.024V):

$$0.158V \times 2^{10} / 1.024V = 158 \quad (5.3)$$

Pela análise da Figura 5.3 é possível verificar que o número de níveis representados está aproximadamente no intervalo de 150 a 160 níveis. Este número está em correspondência com o obtido pelos cálculos efetuados nas expressões anteriores.

É possível verificar também a semelhança de amplitudes entre o sinal transmitido a 1.5 MHz e o de 2 MHz. Isto comprova que os amplificadores e o filtro passa-banda analógico presente antes da ADC tem ganhos de amplitude muito semelhantes para os dois casos. Ou seja, o filtro à partida corresponde de igual forma para as duas bandas de transmissão.

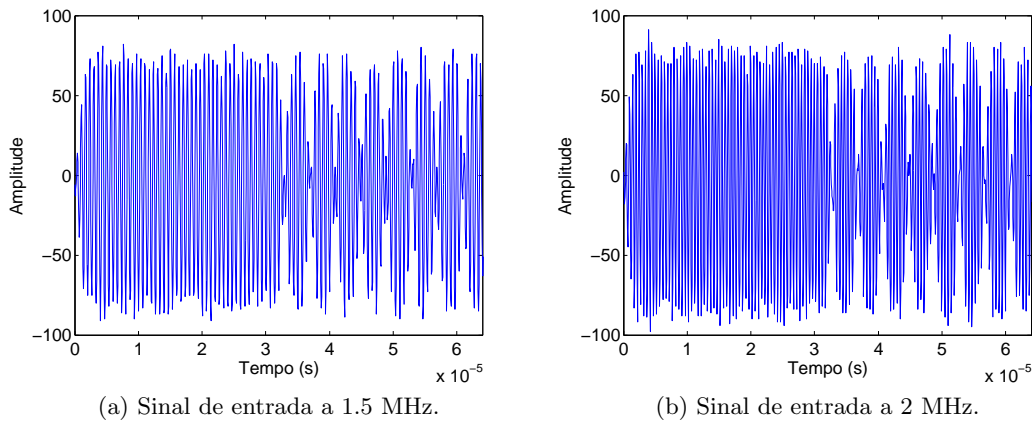


Figura 5.3: Resultados dos sinais de entrada na FPGA a 1.5 MHz e 2 MHz durante o tempo de preâmbulo.

#### 5.2.1.2 Sinal saturado (B)

Como visto anteriormente, este bloco serve essencialmente para normalizar as diferentes amplitudes que podem entrar no desmodulador. A amplitude escolhida para uniformizar (-16 a 15) foi uma amplitude um pouco menor à amplitude de entrada mínima prevista. A Figura 5.4 verifica esta saturação que funciona como previsto.

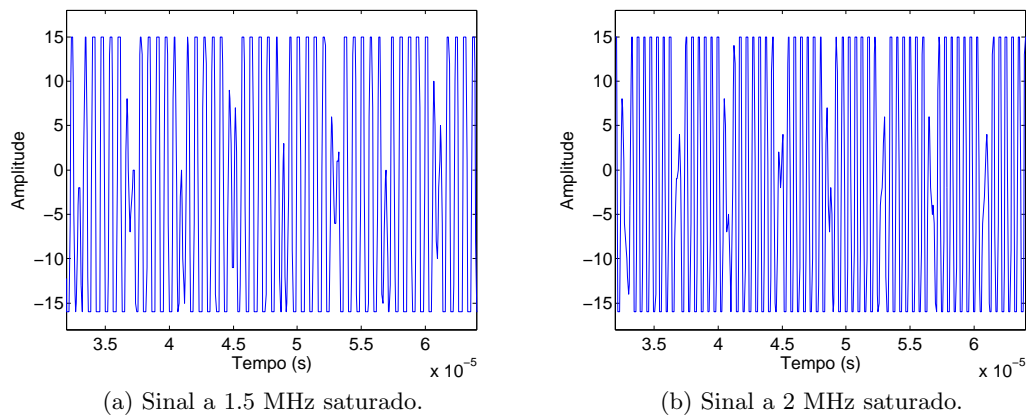


Figura 5.4: Resultados dos sinais a 1.5 MHz e 2 MHz saturados, durante o período de transição de oito 0's no preâmbulo.

#### 5.2.1.3 Sinal depois do filtro passa-banda (C)

A Figura 5.5 mostra os resultados à saída do filtro passa-banda digital, que em cada uma das opções de sub-portadora existe o seu filtro correspondente. Ambos os filtros apresentam resultados muito semelhantes em relação à amplitude, o que facilita ao sistema estar preparado para ambos os casos. Relembre que o ganho da amplitude do erro à entrada do NCO é dependente da amplitude à saída deste filtro.

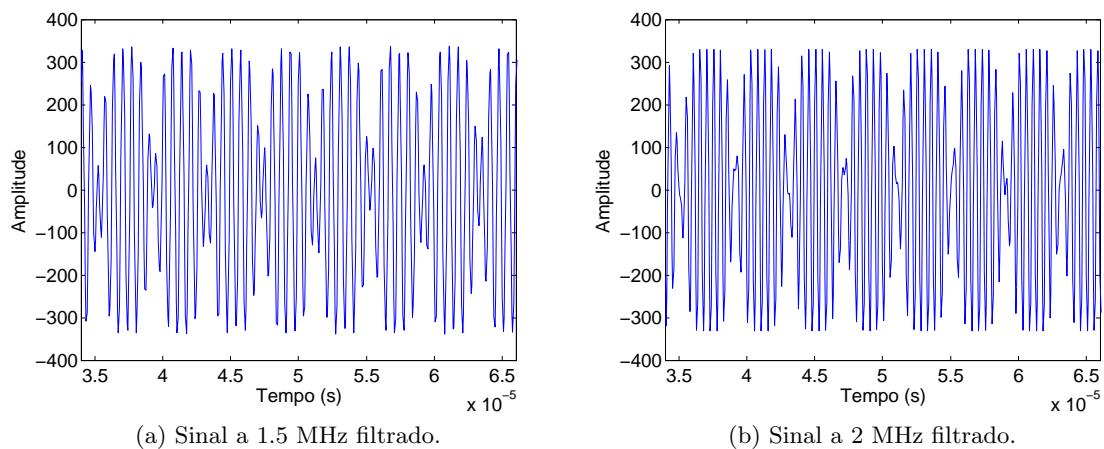


Figura 5.5: Resultados dos sinais a 1.5 MHz e 2 MHz depois do filtro passa-banda correspondente, durante o período de transição de oito 0's no preâmbulo.

### 5.2.2 Costas loop

Mais uma vez, com o objetivo de ser possível indicar cada ponto correspondente ao resultado, a Figura 5.6 relembra o diagrama da implementação do *Costas loop* implementado na FPGA.

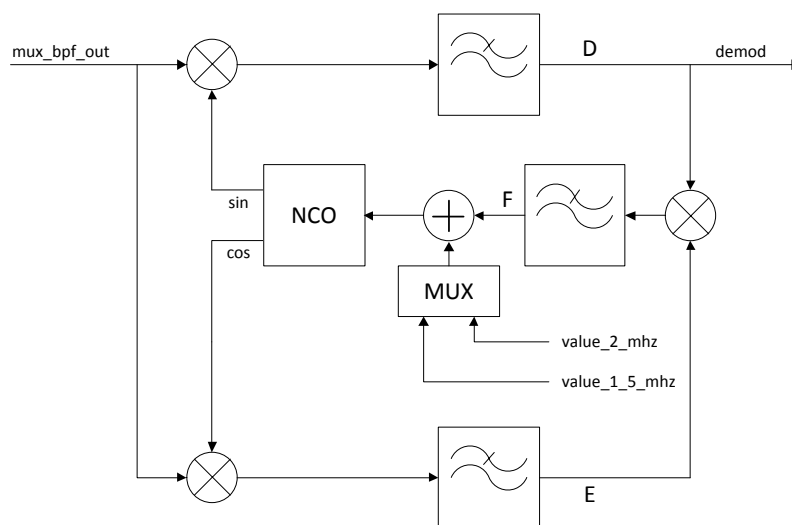


Figura 5.6: Diagrama do *Costas loop* na FPGA.

#### 5.2.2.1 Sinal depois do filtro passa-baixo em fase (D)

A Figura 5.7 apresenta os resultados retirados à saída do *Costas loop*, ou seja, depois do filtro passa-baixo em fase (D na Figura 5.6).

Como é verificado, o sinal apresenta-se desmodulado à frequência de banda-base de 250 kHz. É possível ainda analisar para estes exemplos a duração do tempo de estabilização do *Costas loop*, enquanto a sub-portadora gerada é sincronizada ao sinal recebido.

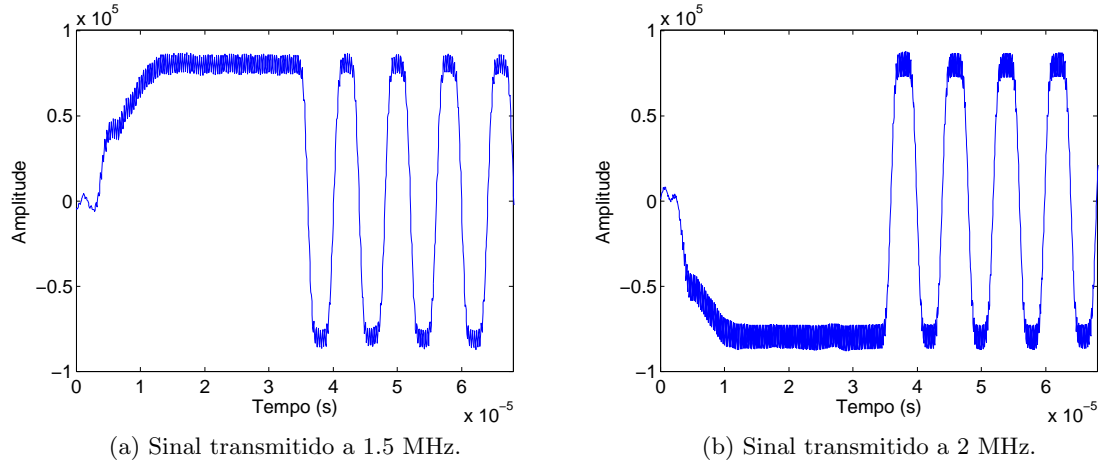


Figura 5.7: Resultados dos sinais transmitidos a 1.5 MHz e 2 MHz desmodulados depois do filtro passa-baixo em fase, durante o tempo do pré-âmbulo.

Nota-se ainda o facto de que no caso do sinal transmitido a 1.5 MHz o *Costas loop* fixa a sub-portadora em fase com o sinal recebido, enquanto que no caso do sinal transmitido a 2 MHz é fixado na fase de  $180^\circ$ . No entanto a fase de fixação do *Costas loop* não tem qualquer ligação com a frequência da sub-portadora, já que o *Costas loop* tanto pode estabilizar em fase como em oposição de fase para qualquer uma das opções da sub-portadora.

#### 5.2.2.2 Sinal depois do filtro passa-baixo em quadratura (E)

Este ramo do *Costas loop*, como explicado anteriormente na sub-secção 3.3.2, tem o objetivo de quando o sinal estiver desfasado com a sub-portadora local, seja reproduzida uma amplitude de erro à entrada do NCO.

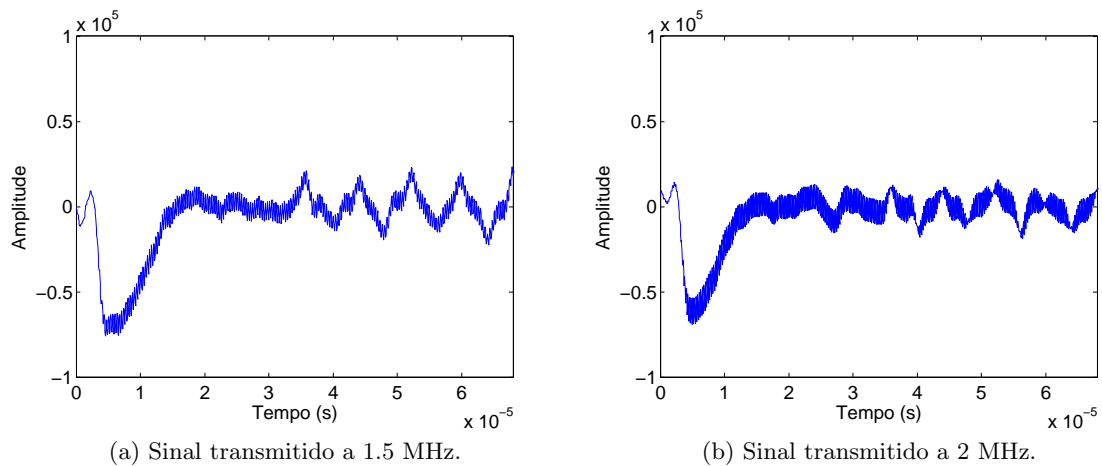


Figura 5.8: Resultados dos sinais transmitidos a 1.5 MHz e 2 MHz depois do filtro passa-baixo em quadratura, durante o tempo do pré-âmbulo.

Ora, o funcionamento deste ramo é visível na Figura 5.8. É possível verificar que existe alguma amplitude durante o tempo inicial mas que rapidamente esta tende para valores próximos de 0.

### 5.2.2.3 Erro depois do filtro passa-baixo dentro da malha (F)

Novamente, de forma a visualizar a estabilização do *Costas loop*, a Figura 5.9 mostra a amplitude do erro à entrada do NCO, ou seja, logo a seguir à saída do filtro passa-banda dentro da malha.

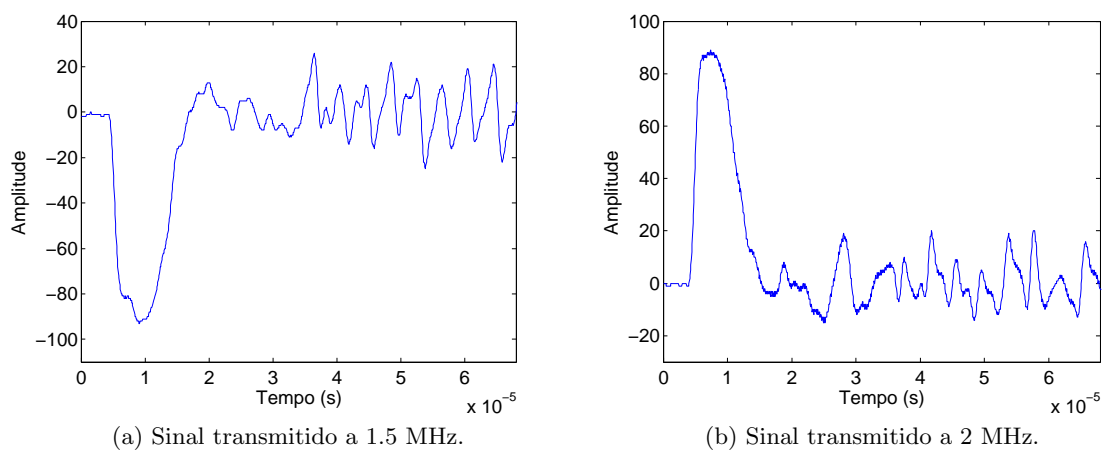


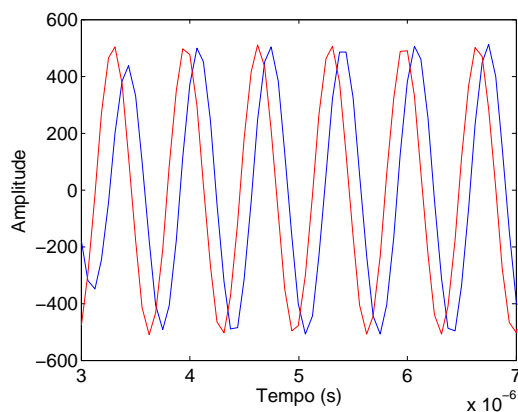
Figura 5.9: Resultados da amplitude do erro à entrada do NCO de correção da sub-portadora gerada face aos sinais de entrada a 1.5 MHz e a 2 MHz, durante o tempo do pré-âmbulo.

Medindo o tempo de estabilização nos dois casos, este é de aproximadamente  $10 \mu s$ , o que corresponde ao objetivo traçado inicialmente de conseguir que a estabilização ocorresse dentro do tempo máximo de sub-portadora pura do pré-âmbulo.

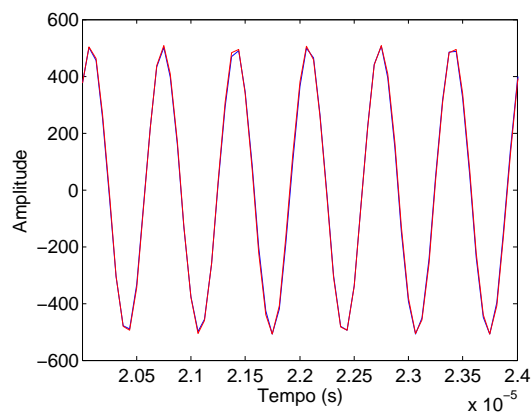
### 5.2.2.4 Sincronização entre o sinal recebido e a sub-portadora à saída do NCO (sin)

De forma a exemplificar melhor a sincronização entre o sinal de entrada no *Costas loop* e a sub-portadora gerada localmente na FPGA, as Figuras 5.10 e 5.11 apresentam os resultados destes dois sinais sobrepostos.

Como analisado anteriormente, verifica-se que nos resultados retirados do sinal a 1.5 MHz, a sub-portadora é sincronizada em fase com o sinal recebido, enquanto que nos resultados do sinal a 2 MHz, a sub-portadora é sincronizada a  $180^\circ$  do sinal recebido. Isto em nada prejudica o sistema, pois a decodificação dos símbolos transmitidos é efetuada com base na ocorrência de transições de fase. Note-se que a sincronização em fase ou em oposição de fase é um mero acaso em qualquer obtenção de resultados e não tem qualquer ligação à frequência de transmissão da sub-portadora.

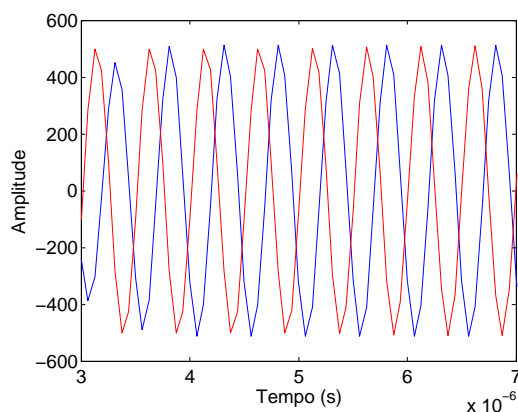


(a) Início do préâmbulo.

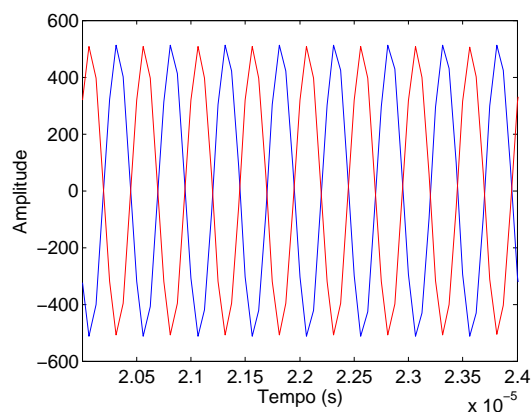


(b) Préâmbulo depois dos sinais estarem síncronos.

Figura 5.10: Sincronização entre o sinal recebido e a sub-portadora à saída do NCO no sinal transmitido a 1.5 MHz. A vermelho, sub-portadora gerada pelo NCO. A azul, sinal recebido.



(a) Início do préâmbulo.



(b) Préâmbulo depois dos sinais estarem síncronos.

Figura 5.11: Sincronização entre o sinal recebido e a sub-portadora à saída do NCO no sinal transmitido a 2 MHz. A vermelho, sub-portadora gerada pelo NCO. A azul, sinal recebido.

### 5.2.3 Recuperação do relógio de transmissão em banda-base

Os resultados retirados da recuperação do relógio de transmissão são visíveis na Figura 5.12. Esta figura mostra os três integradores e o instante em que o sinal foi amostrado. O período de tempo visível é, mais uma vez, o tempo de préâmbulo no início da trama.

Este resultado representa praticamente o pior caso previsto, já que os instantes de amostragem no início das primeiras transições estão muito próximos da origem. É possível verificar que o relógio recupera totalmente durante o período de transição dos oito 0's iniciais do préâmbulo, pois no fim os instantes de amostragem encontram-se nos picos do integrador central.



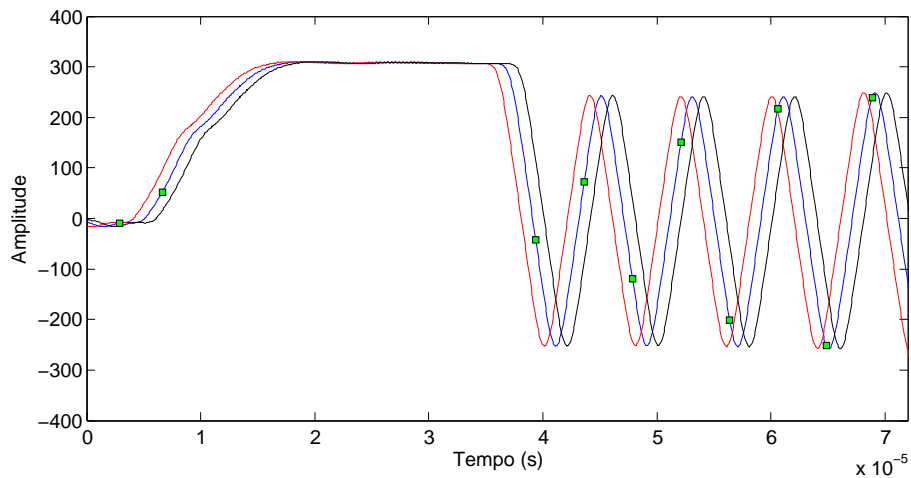


Figura 5.12: Integração do preâmbulo do sinal a 1.5 MHz desmodulado. São visíveis os três integradores em que o integrador central é o relativo à amostragem. Os pontos correspondem aos instantes de amostragem.

### 5.3 Testes de avaliação e validação através do osciloscópio e sem ruído adicionado

A constituição desta segunda montagem teve como objetivo analisar o resultado da desmodulação com o auxílio do osciloscópio em vez de usar a ferramenta *ChipScope*. A vantagem passaria por conseguir sobrepor vários resultados aleatórios e assim conseguir obter conclusões mais abrangentes em relação ao comportamento do sistema. A Figura 5.13 representa o esquema usado.

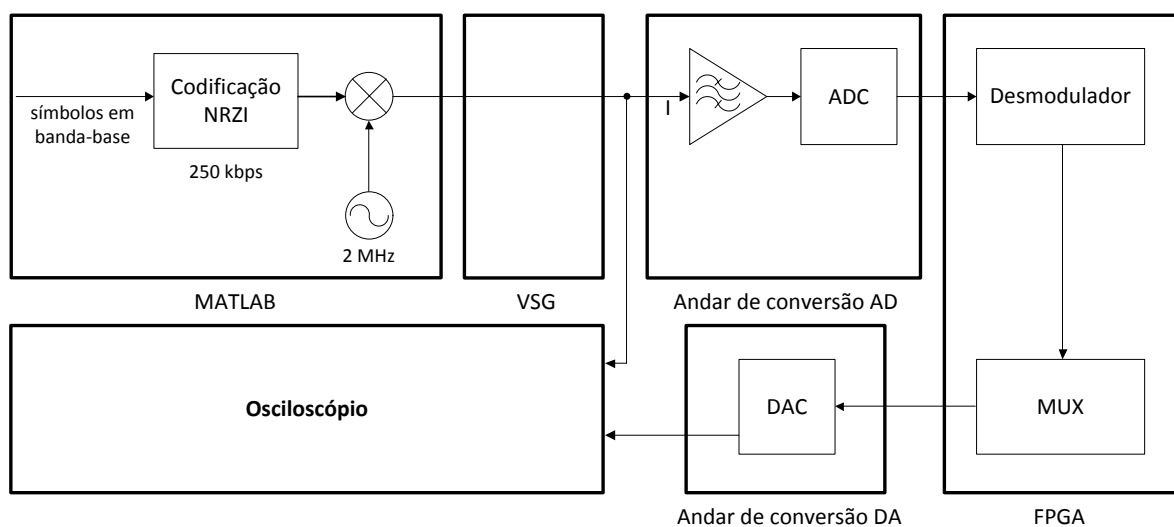


Figura 5.13: Diagrama de blocos da montagem para testes através do osciloscópio e sem ruído adicionado.

Agora, o sinal gerado no MATLAB já contém a sub-portadora (apenas vai ser analisado a opção de sub-portadora de 2 MHz), e assim o VSG tem o único propósito de converter o sinal digital para analógico. Neste demonstrador não foi introduzido ruído ao sinal gerado.

Para conseguir visualizar os sinais no osciloscópio foi utilizada uma placa, que continha uma DAC de 12 *bits*, e na FPGA, através de um *multiplexer*, era possível escolher dos vários sinais internos do desmodulador o qual a transmitir para a DAC.

### 5.3.1 Sinal de entrada na FPGA

A Figura 5.14 representa a comparação entre o sinal à entrada do andar de conversão AD e o sinal obtido à entrada da FPGA, ou seja, depois da conversão da ADC.

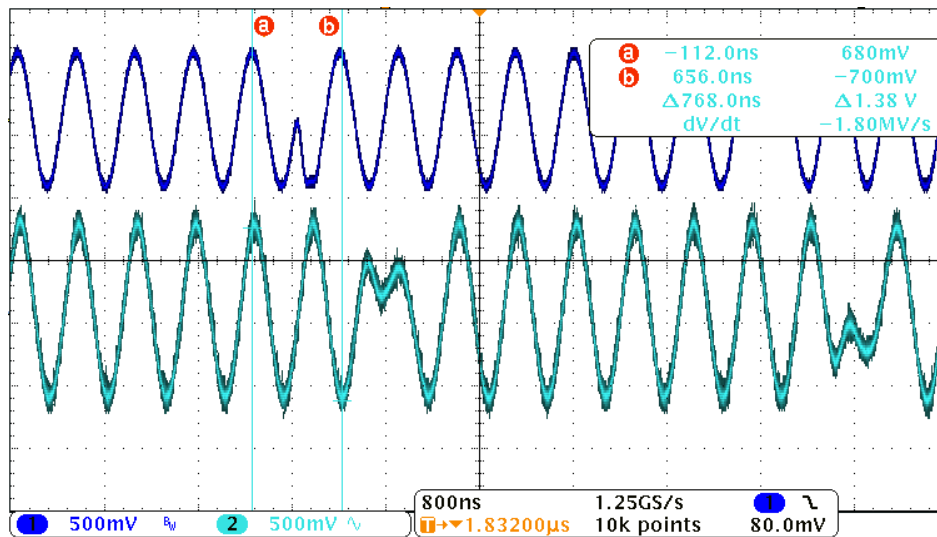


Figura 5.14: Sinal de entrada na FPGA a 2 MHz durante uma transição de fase (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima).

O período quantificado de 768 ns (medido através da comparação entre os inícios de transição de fase nos dois sinais) é o atraso medido entre os dois sinais. Este atraso é o resultado dos atrasos de conversão na ADC, dos diversos *flip-flops* usados na FPGA e ainda da DAC usada para visualizar o sinal no osciloscópio. Este período corresponde a 12 ciclos do relógio de 16 MHz.

### 5.3.2 Sinal de correção do *Costas loop* (F)

Na Figura 5.15 é possível analisar o resultado de várias respostas de correção ao NCO do *Costas loop* (recordar o ponto na Figura 5.6) com persistência infinita, isto é, estão sobrepostos todos os resultados obtidos. É possível concluir, através de inúmeros testes, que a correção nunca ultrapassou o tempo de preâmbulo relativo à transmissão de sub-portadora pura (32 a 36  $\mu$ s). Assim, para todos os casos, quando a parte de transmissão de oito 0's inicia, a sub-portadora gerada pelo NCO já se encontra síncrona com o sinal recebido.

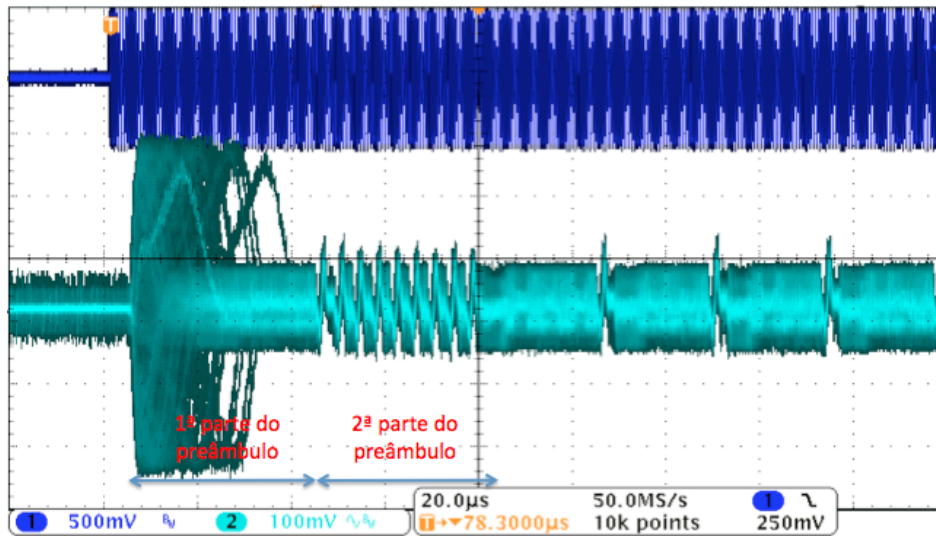


Figura 5.15: Sinal de correção do *Costas loop* à entrada do NCO (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima).

### 5.3.3 Sinal desmodulado à saída do *Costas loop* (D)

A Figura 5.16 apresenta os vários resultados obtidos do sinal desmodulado à saída do *Costas loop* (recordar o ponto na Figura 5.6), mais uma vez com recurso a persistência infinita. Novamente, é possível analisar que a amplitude do sinal desmodulado torna-se máxima antes de começar a parte do préambulo de transição de oito 0's.

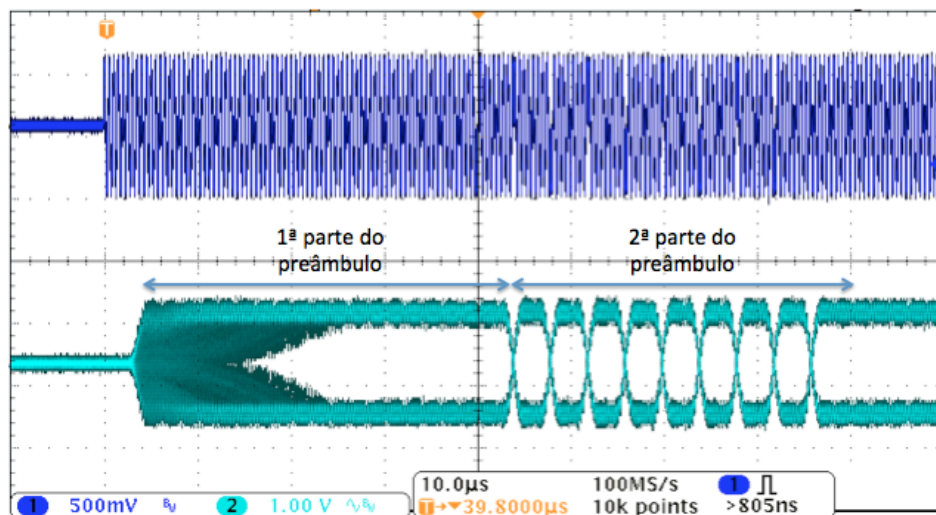


Figura 5.16: Sinal desmodulado à saída do *Costas loop* (em baixo) em comparação com o sinal à entrada do andar de conversão AD (em cima).

### 5.3.4 Recuperação relógio banda-base

Na Figura 5.17 é visível o sinal amostrado e decodificado de NRZI, assim como o *trigger* relativo à amostragem de 250 kHz do mesmo, durante o período do préambulo inicial.

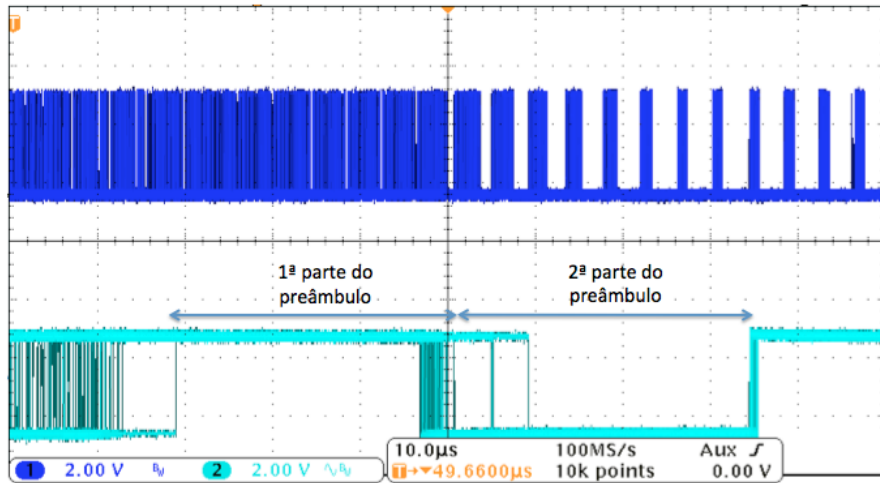


Figura 5.17: Sinal amostrado e decodificado de NRZI (em baixo) e o trigger relativo à sua amostragem (em cima).

É possível verificar que assim que termina a transmissão dos oito 0's iniciais, o relógio encontra-se completamente recuperado. Essa recuperação demora sensivelmente dois ou três 0's. Isto é comprovado através da figura, pois durante esses primeiros 0's o sinal amostrado por vezes não corresponde ao nível certo.

Também é possível analisar o *jitter* do *trigger* correspondente ao relógio de 250 kHz. Este *jitter* vai diminuindo no decorrer do período do preâmbulo.

### 5.3.5 Diagrama de olho do sinal desmodulado

A Figura 5.18 representa o diagrama de olho do sinal desmodulado à saída do *Costas loop*, tal como o *trigger* relativo à recuperação do relógio de 250 kHz. Este resultado foi retirado sensivelmente a meio da parte do preâmbulo de transmissão dos oito 0's (aos 50 µs).

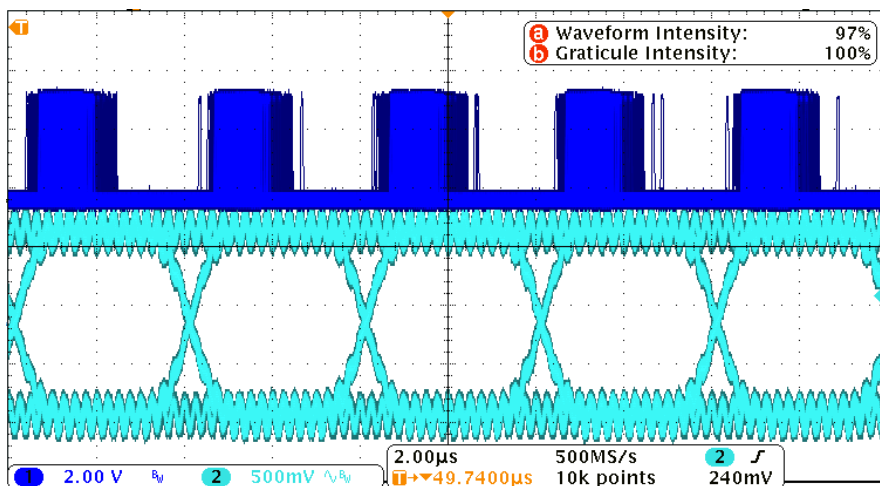


Figura 5.18: Diagrama de olho do sinal desmodulado (em baixo) e análise do jitter do trigger relativo à sua amostragem (em cima).

Como previsto, nesta fase o *Costas loop* já está completamente síncrono, facto que é comprovado pelo diagrama de olho da figura. Quanto ao relógio de 250 kHz, embora o preâmbulo ainda não tenha terminado, e por isso o *jitter* do *trigger* ser bastante elevado, já são notórios e distinguíveis os instantes de amostragem do sinal.

De notar que o instante de amostragem ideal seria no fim do símbolo. O desfasamento observado entre o relógio e o sinal desmodulado deve-se ao atrasado provocado pelos integradores.

## 5.4 Testes de avaliação e validação através do osciloscópio e com ruído adicionado

Esta terceira e última montagem teve como objetivo analisar o resultado da desmodulação do sinal com ruído com o auxílio do osciloscópio. Novamente, a vantagem passaria por conseguir sobrepor vários resultados aleatórios, observando agora o comportamento do sistema com o ruído aplicado. A Figura 5.19 representa o esquema usado.

Foi utilizado um gerador de sinais adicional como gerador de ruído gaussiano branco. A amplitude do ruído a gerar foi determinada através de um *Spectrum Analyser*. Fixou-se a amplitude correspondente à potência necessária de -24.5 dBm numa banda de 500 kHz quando o mesmo foi observado no *Spectrum Analyser*. Esta potência é 18 dB abaixo da potência do sinal de entrada, correspondendo assim a um SNR de 18 dB. Relembre-se que o SNR de 18 dB era o pior caso previsto na secção 4.2.

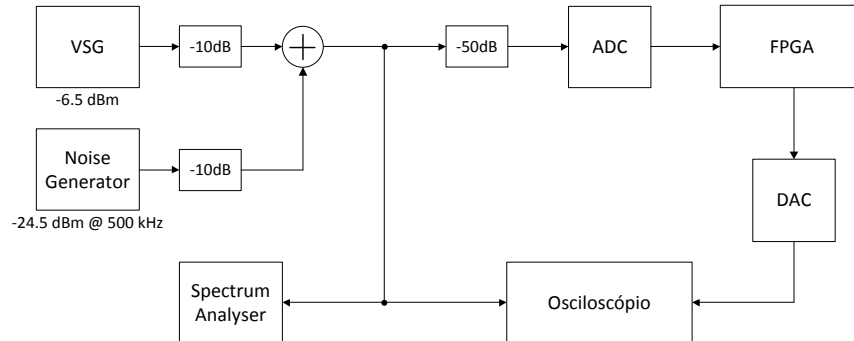


Figura 5.19: Diagrama de blocos da montagem para testes através do osciloscópio e com ruído adicionado.

Os atenuadores tinham como propósito a obtenção de um sinal à entrada do andar de conversão AD equivalente ao sinal previsto num sistema real de pior caso (potência mínima de -67 dBm). Com a potência do sinal gerado de -6.5 dBm e 60 dB de atenuação, o sinal à entrada do andar de conversão AD tinha uma potência de cerca de -66.5 dBm.

### 5.4.1 Sub-portadora recuperada (“sin”)

A Figura 5.20 representa a sub-portadora gerada pelo *Costas loop* (recordar o ponto na Figura 5.6) com persistência infinita (sobrepostas as sub-portadoras geradas em fase e em oposição de fase). Este resultado foi retirado no fim do preâmbulo de transmissão (aos 64  $\mu$ s).

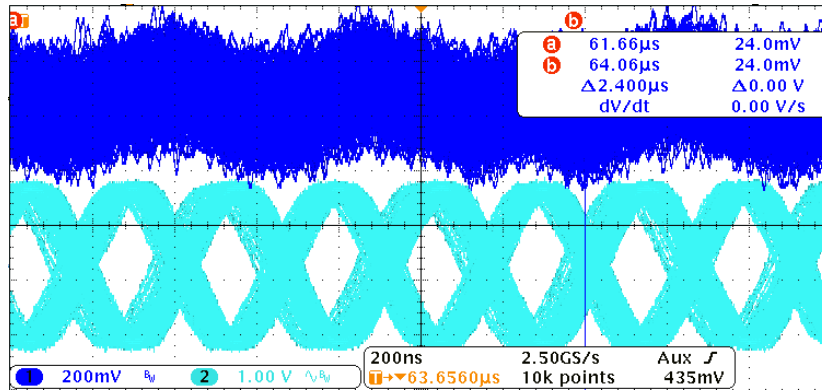


Figura 5.20: Sub-portadora gerada pelo *Costas loop* (em baixo) de um sinal de entrada com SNR de 18 dB (em cima).

É possível verificar que com um SNR de 18 dB a sub-portadora gerada apresenta algum *jitter*, embora se verifique que o *Costas loop* continua a sincronizar com o sinal de entrada. É notória a diferença e a separação entre períodos pelo que é uma sub-portadora com perspectivas de poder ser multiplicado pelo sinal de entrada e desmodular com alguma eficácia.

#### 5.4.2 Sinal de correção do *Costas loop* (F)

Na Figura 5.21 é possível analisar novamente o resultado de várias respostas de correção ao NCO do *Costas loop* (recordar o ponto na Figura 5.6) com persistência infinita. O início do preâmbulo corresponde ao instante de tempo assinalado pela linha vertical azul à esquerda.

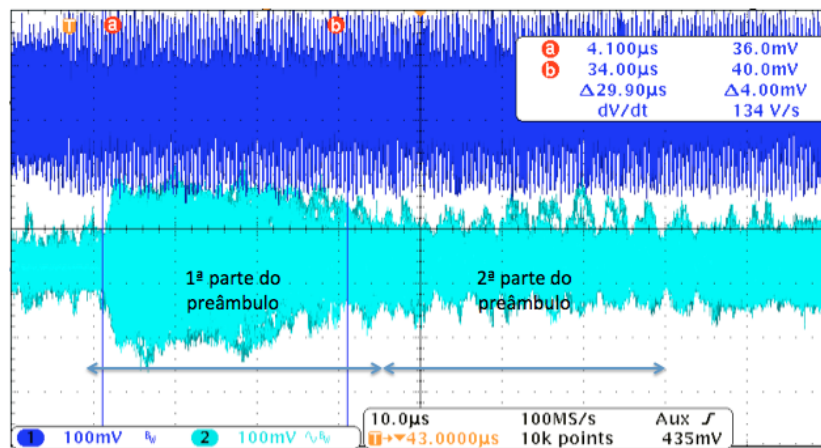


Figura 5.21: Sinal de correção do *Costas loop* à entrada do NCO (em baixo) em comparação com um sinal à entrada do sistema com SNR de 18 dB (em cima).

Desta vez não é simples determinar o tempo de estabilização do *Costas loop*, embora na Figura 5.20 se tenha observado uma sub-portadora estável no fim período do preâmbulo, o que leva a concluir que o *Costas loop* estabiliza a partir deste período.

Mesmo assim, note-se a mancha de maior amplitude no início de preâmbulo que diminui durante o decorrer do preâmbulo, efetivamente no fim da 1ª parte do preâmbulo. É mais um fator a comprovar a estabilização do sistema.

### 5.4.3 Integrador e amostrador do sinal

A Figura 5.22 apresenta o resultado da integração do sinal desmodulado após o *Costas loop* e os símbolos em banda-base após a amostragem e decodificação NRZI.

Este resultado foi retirado durante um período onde só eram transmitidos 0's. No entanto, é visível que durante o período em questão ocorreu um erro, já que foi amostrado um símbolo '1'.

A ocorrência destes erros foi verificada mais vezes e depois analisada com algum detalhe. Foi verificado que, nestes casos, no instante de amostragem correspondente a amplitude do integrador estava demasiado próximo do limiar de decisão. Por vezes, o sinal integrado estava mesmo com sinal contrário ao que era suposto ser amostrado (isto é, era negativo em vez de positivo ou vice-versa).

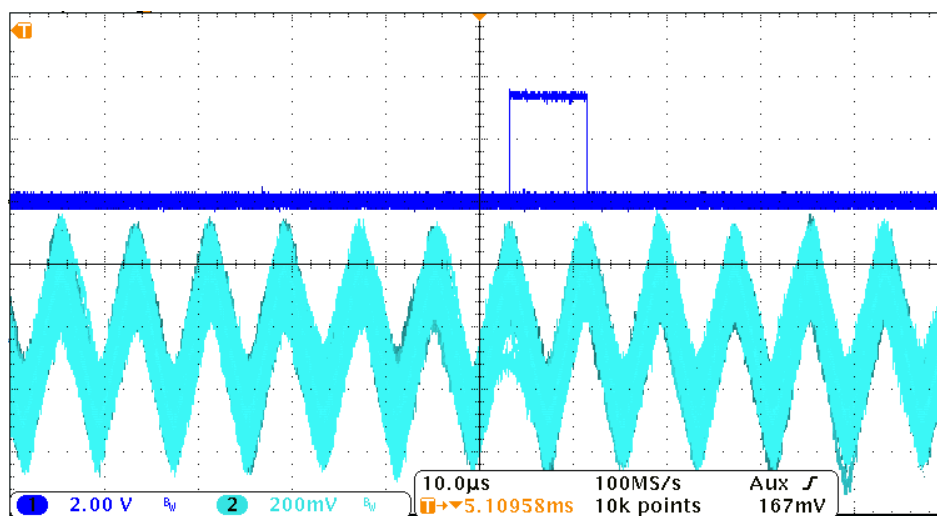


Figura 5.22: Sinal integrado do sinal desmodulado (em baixo) confrontado com o sinal amostrado e decodificado de NRZI (em cima) com um sinal à entrada do sistema com SNR de 18 dB.

Os erros observados podem ser provocados por dois fatores principais:

- A amplitude do pico não ser tão alta como a desejada, estando demasiado próxima do nível de decisão e assim com mais probabilidade da amostragem não ocorrer sobre o valor pretendido. Esta diminuição de amplitude pode ter sido provocada por algum momento de dessincronização do *Costas loop*;
- A correção do relógio de amostragem de 250 kHz ser demasiado ativa (quando há erro corrige 4 amostras de uma só vez). Se o relógio for corrigido em instantes não necessários tem como consequência a dessincronização da amostragem. Depois a amostragem pode ocorrer no instante de valor com sinal contrário ao pretendido.

Este problema pode ser atenuado se a correção mais ativa ocorresse apenas durante o preâmbulo inicial de forma a sincronizar o relógio de banda-base rapidamente. Depois durante a amostragem da trama de informação, o relógio tinha uma resposta de correção mais lenta, de forma a evitar correções não pretendidas, contribuindo para a estabilidade do sistema.





## Capítulo 6

# Conclusões e Trabalho Futuro

### 6.1 Conclusões

O principal objetivo deste trabalho foi a construção de um recetor para comunicações DSRC baseado em tecnologia de *Software Defined Radio (SDR)*.

Depois de uma breve introdução sobre o estado e importância dos sistemas ITS na evolução tecnológica da sociedade de hoje, procedeu-se a uma abordagem da norma EN12253 que especifica a camada física das comunicações DSRC. Partindo desta referência, introduziu-se o conceito de SDR, apresentando as suas vantagens e interligando-as com a utilidade das mesmas em DSRC. Essas vantagens passariam não só pela simplificação e redução de custos do sistema, como também pela adição da possibilidade de atualização e maior flexibilidade de adaptação. Foram apresentadas ainda as possíveis configurações de SDR e por fim os componentes importantes a utilizar numa aplicação DSRC com recurso a SDR.

Com vista à conceptualização do sistema, foi estudada a modulação BPSK e as suas formas de receção. Depois de comparados os vários métodos de desmodulação, através de simulações do processo completo de *Uplink*, optou-se pela utilização do método *Costas loop*.

De seguida, o sistema foi implementado de forma gradual. Foi necessário construir uma placa para a conceção dos dois sinais recebidos (em fase e em quadratura) à frequência da sub-portadora. Esta conceção visava a amplificação dos sinais para níveis de amplitude aptos a serem lidos pela ADC, filtragem inicial da banda pretendida e a conversão analógico-digital de modo a que, de seguida, os sinais fossem processados na FPGA. Na FPGA foram implementados a desmodulação *Costas loop*, para recuperar a sub-portadora e transladar o sinal para banda-base, e ainda a recuperação de relógio em banda-base, para depois amostrar e descodificar os símbolos recebidos.

Através de testes de validação e avaliação do sistema, retiraram-se resultados e verificou-se que foi possível implementar o sistema, pois produziram resultados muito semelhantes aos previstos pelas simulações. No entanto, quando adicionado ruído ao sinal recebido, foi visível a presença de alguns erros na amostragem dos símbolos. Foram detetados algumas das causas para a ocorrência destes erros, como também foram sugeridas algumas alternativas específicas para melhorar.

Como conclusão, é possível afirmar que a parte constituinte do RSU aqui implementada está no caminho desejado para a evolução de equipamentos de comunicações DSRC. É assim de esperar que o trabalho aqui apresentado seja útil e funcional para uma possível inclusão no futuro.

## 6.2 Trabalho Futuro

Como trabalho futuro, sugerem-se as seguintes indicações para os dois principais blocos implementados:

**Andar de Amplificação/Filtragem e Conversão AD** Nesta placa, com vista à continuidade da simplificação do sistema e da redução de custos, uma das seguintes possibilidades poderia ser o caminho a seguir numa futura evolução:

- Eliminar a amplificação/filtragem analógica. Uma das formas de evitar nesta fase a necessidade de amplificação dos sinais era reduzir significativamente a amplitude de referência das tensões de entrada na ADC. Assim, estaria-se a aproveitar os níveis da ADC para níveis de tensão muito menores, substituindo o efeito da amplificação inicial. Por sua vez, poderia-se remeter a filtragem apenas para a FPGA.
- Não utilizar ADC. Por se tratar de uma receção de símbolos modulados em fase, a amplitude do sinal não tem qualquer influência na desmodulação. Assim, pode ser estudada a hipótese de amplificar bastante o sinal recebido, garantindo a saturação da amplitude mínima esperada do sinal. Com um sinal de apenas dois níveis, este era transferido para a FPGA através de uma entrada digital.

**Implementação do Desmodulador na FPGA** Algumas otimizações que poderiam ser efetuadas na implementação da FPGA:

- Uma forma de resolver o problema do bloco de atraso de  $90^\circ$  para a opção de 1.5 MHz, presente no início do desmodulador (Sub-seção 4.4.2), era usar dois relógios distintos para cada uma das opções de sub-portadora a receber. Isto levaria a que alguns dos filtros do *Costas loop* tivessem que conter dois grupos de coeficientes (um para a opção de 1.5 MHz e outro para a opção de 2 MHz).
- Reduzir a ordem dos filtros usados, de forma a não estar dependente de FPGAs de topo e contribuindo assim para a redução de custos. Uma forma de aproveitar os recursos da FPGA era utilizar a funcionalidade de *multiplexar* o mesmo filtro por vários sinais, em vez de usar filtros iguais em ramos distintos. Pode ser também estudada a possibilidade de aproximar os coeficientes dos filtros a valores  $2^n$  para que os multiplicadores se resumissem a *shifts lefts*.
- Para evitar a ocorrência de alguns dos erros observados no recuperador de relógio de banda-base, são aconselhadas uma das duas alternativas:

Tornar o recuperador implementado mais adaptável. Durante o preâmbulo, deixar que o recuperador tenha uma correção rápida e abrupta, mas depois durante a trama de informação tornar a correção mais leve.

Implementar outro tipo de recuperador baseado nos recuperadores *Clock and Data Recoverys (CDRs)*. Estes recuperadores utilizam uma PLL. A principal vantagem passaria pela utilização do bloco de detetor de fase que deteta a dessincronização entre o relógio recuperado e o sinal a amostrar. A própria amplitude dos picos produzidos pelo detetor serve corrigiria o desfasamento do relógio. Deste tipo de recuperadores os mais conhecidos são o *Alexander (Bang-Band)* e o *Hogge*.

## Apêndices



## Apêndice A

# Placa Construída (PCB e Esquema Elétrico)

Neste Apêndice são apresentadas as figuras do esquema elétrico (Figura A.1) e do *layout* do PCB à escala 1:1 (Figura A.3) da placa implementada para amplificação, filtragem e conversão AD dos sinais recebidos a serem desmodulados pela FPGA.



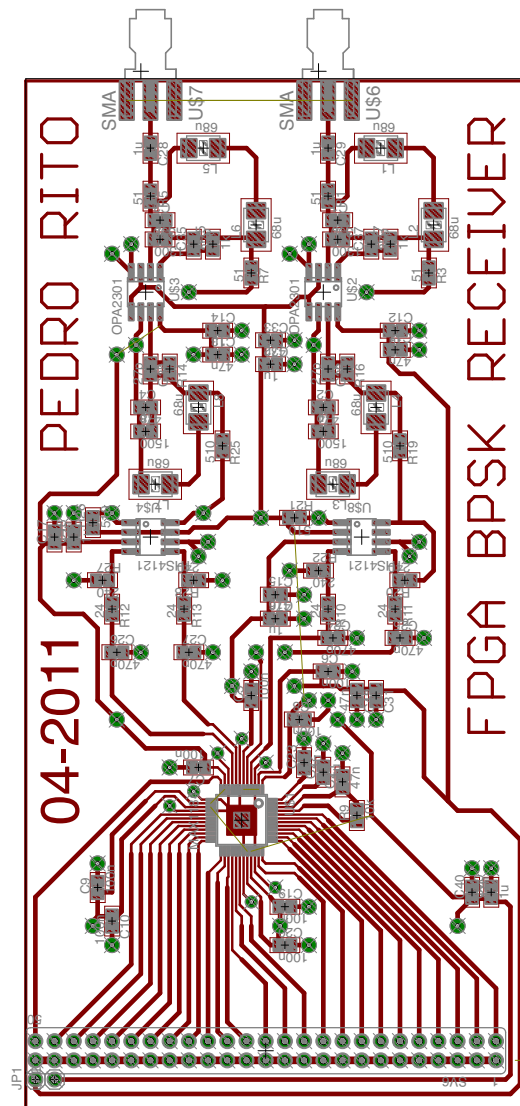


Figura A.2: Layout PCB à escala 1:1 da placa implementada de amplificação, filtragem e conversão AD com representação dos componentes.

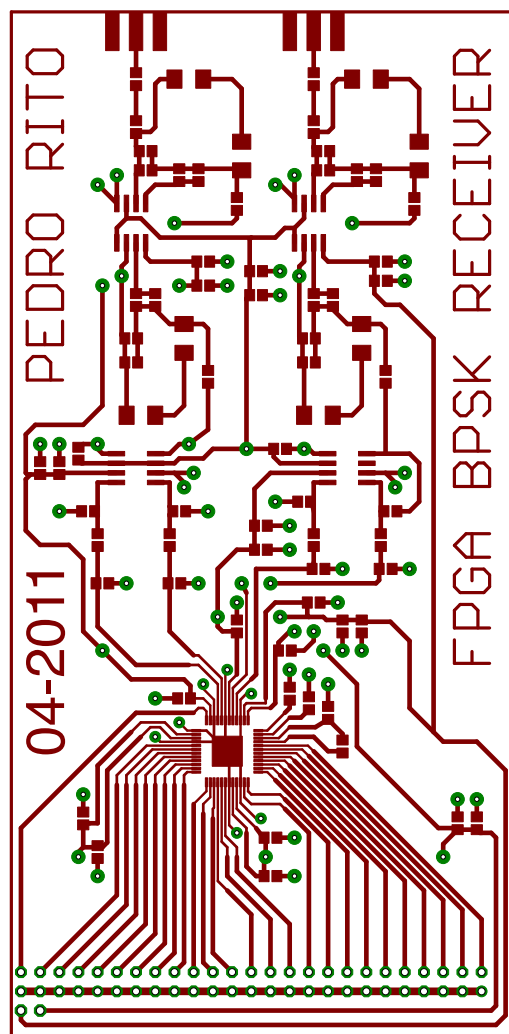


Figura A.3: Layout PCB à escala 1:1 da placa implementada de amplificação, filtragem e conversão AD.



## Apêndice B

# Esquema da implementação na FPGA

Neste Apêndice é apresentada a figura do esquema completo da implementação efetuada na FPGA (Figura B.1).

Este esquema auxilia a interpretação da dissertação durante a secção 4.4, pois apresenta uma visão completa de todo o projeto do desmodulador e decodificador desenvolvidos através da FPGA.

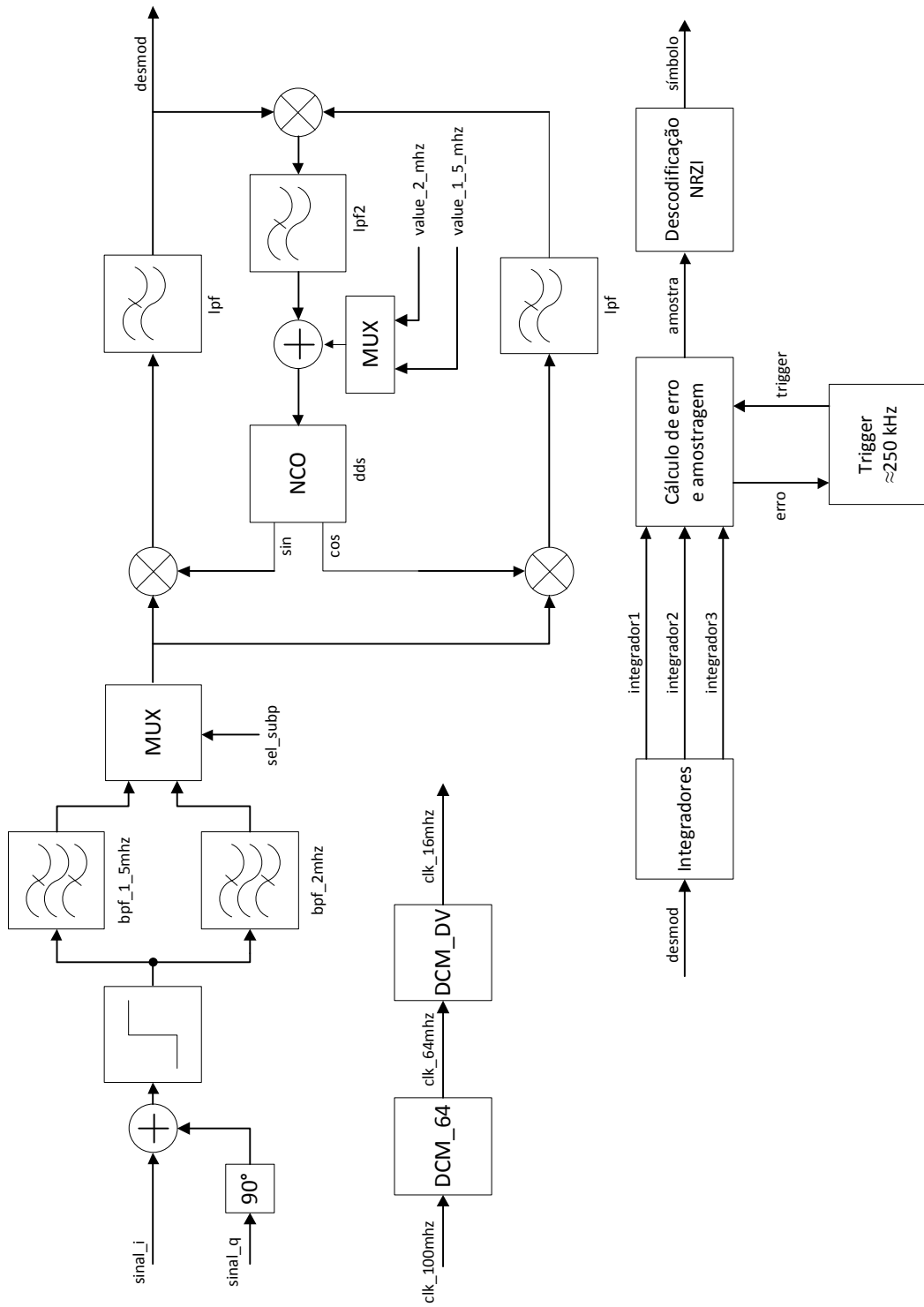


Figura B.1: Esquema completo da implementação efetuada na FPGA. Todos os blocos (exceto os DCMs) são síncronos pelo relógio de 16 MHz (clk\_16mhz) produzido pelo DCM\_DV.

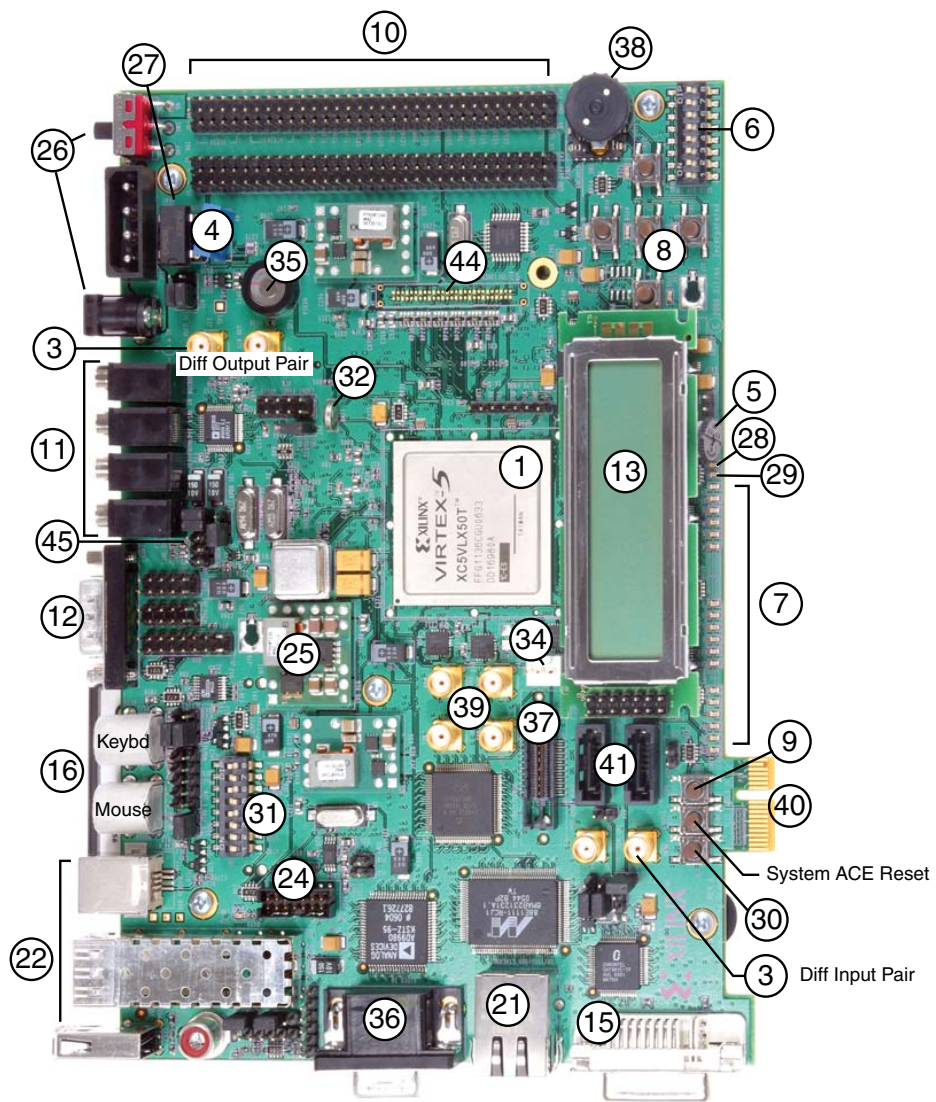
## Apêndice C

# Placa de Desenvolvimento Xilinx ML505

Neste Apêndice são apresentadas fotografias da placa de desenvolvimento usada para a implementação do projeto na FPGA. Na Figura C.1 é possível visualizar a fotografia da mesma com enumeração dos vários módulos presentes na placa. De forma a compreender as secções essenciais da placa, isto é, as que foram usadas na realização do projeto, é apresentado de seguida a legenda respetiva:

- **1** - FPGA utilizada na implementação, uma Xilinx Virtex-5.
- **6** - *DIP-switches*. Utilizados para configurar algumas opções, como a escolha do tipo de sub-portadora a receber (1.5 MHz ou 2 MHz), a escolha de qual o sinal a enviar para a DAC na fase de resultados no demonstrador 3 e *enable* do *reset*.
- **7** - LEDs. Utilizados na fase inicial de implementação para *debug* do projeto.
- **10** - *Headers* de expansão utilizados para a interface entre placa construída de amplificação, filtragem e conversão AD dos sinais recebidos e a FPGA.

Na Figura C.2 é possível visualizar a placa desenvolvimento ML505 em conjunto com a placa construída de amplificação, filtragem e conversão analógico-digital. É assim possível compreender a forma como estas duas placas eram interligadas, através dos *headers* de expansão da placa ML505.



UG347\_01\_102907

Figura C.1: Fotografia da placa de desenvolvimento da Xilinx ML505 com enumeração de secções.



Figura C.2: Fotografia da placa de desenvolvimento da Xilinx ML505 com a placa construída de amplificação, filtragem e conversão analógico-digital.



# Bibliografia

- [Com04] Comité Européen de Normalisation (CEN). *European Standard EN 12253*, Julho 2004.
- [Cos02] John P. Costas. Synchronous communications. *Proceedings of the IEEE*, 90(8), Agosto 2002.
- [Hay00] Simon Haykin. *Communication Systems*. Wiley, 4th edition, Maio 2000.
- [IEE11] IEEE. IEEE DYSpan Standards Committee. <http://grouper.ieee.org/groups/>, 2011.
- [III00] J. Mitola III. *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Department of Teleinformatics, Royal Institute of Technology (KTH), Stockholm, Sweden, Maio 2000.
- [Kes05] Walt Kester, editor. *The Data Conversion Handbook*. Newnes, an imprint of Elsevier, 2005.
- [Mad01] Nuno Borges Carvalho; Raquel Castro Madureira. Intermodulation interference in the gsm/umts bands. In *III Conferência de Telecomunicações*, Figueira da Foz, Abril 2001.
- [Man07] John G. Proakis; Dimitris G. Manolakis. *Digital Signal Processing*. Prentice Hall, 2007.
- [MAX06] MAXIM. *MAX1184, Dual 10-Bit, 20MSPS, 3V, Low-Power ADC with Internal Reference and Parallel Outputs*, Julho 2006.
- [Mit95] J. Mitola. The Software Radio Architecture. *IEEE Communications Magazine*, 33(5):26 – 38, 1995.
- [Ree02] Jeffrey H. Reed. *Software Radio*. Prentice Hall, 2002.
- [Sch06] M. Schwartz. *Information Transmission, Modulation and Noise*. McGraw-Hill Publishing Company, 4th edition, 2006.
- [Smi07] A. Sedra; K. C. Smith. *Microelectronic Circuits*. Oxford University Press, 5th edition, 2007.
- [Tex04] Texas Instruments. *THS4121, High-Speed Fully Differential I/O Amplifiers*, Outubro 2004.

- [Tex07] Texas Instruments. *OPA2301, Low-Noise, High-Speed, 16-Bit Accurate, CMOS Operational Amplifier*, Junho 2007.
- [Xil09] Xilinx. *ML505/ML506/ML507 Evaluation Platform - User Guide*, Outubro 2009.
- [Xil10a] Xilinx. *LogiCORE IP DDS Compiler v4.0*, Abril 2010.
- [Xil10b] Xilinx. *LogiCORE IP FIR Compiler v5.0*, Abril 2010.